UNIVERSITÉ DE MONTRÉAL

EMPIRICAL MEANS TO VALIDATE SKILLS MODELS AND ASSESS THE FIT OF A STUDENT MODEL

BEHZAD BEHESHTI
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
DÉCEMBRE 2015

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

EMPIRICAL MEANS TO VALIDATE SKILLS MODELS AND ASSESS THE FIT OF A STUDENT MODEL

présentée par : BEHESHTI Behzad
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. GUEHENEUC Yann-Gael, Doctorat, président
M. DESMARAIS Michel C., Ph. D., membre et directeur de recherche
M. GAGNON Michel, Ph. D., membre
M. BRAULT Jean-Jules, Ph. D., membre
M. PARDOS Zachary A., Ph. D., membre externe

DEDICATION

To my beloved family

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Michel C. Desmarais who has provided constant guidance and encouragement throughout my research at Ecole Polytechnique de Montreal. I do not know where my research would go without his patience and efforts.

I would like to thank administrative staff and system administrators in the department of Computer Engineering at Ecole Polytechnique de Montreal for their incredible helps.

At the end, words cannot express how grateful I am to my family who never stopped supporting me even from distance. A special thanks to my friends who always inspire me to strive towards my goals.

RÉSUMÉ

ABSTRACT

In educational data mining, or in data mining in general, analysts that wish to build a classification or a regression model over new and unknown data are faced with a very wide span of choices. Machine learning techniques nowadays offer the possibility to learn and train a large and an ever growing variety of models from data. Along with this increased display of models that can be defined and trained from data, comes the question of deciding which are the most representative of the underlying ground truth. The main objective of this thesis is assessing model fit for student test results.

Assessing whether a student model is a good fit to the data is non trivial. The standard practice is to train different models, and consider the one with the highest predictive performance as the best fit. But each model may involve different machine learning algorithms that carry their own set of parameters and constraints imposed on the corresponding model. This results in a large space in which to explore model performance. The actual best fitting model may have been overlooked due to an unfortunate choice of the algorithm's parameters. Therefore, the best performer may not be the model that is most representative of the ground truth, but instead it may be the result of contextual factors that make this model outperform the ground truth one.

We investigate the question of assessing different model fits using synthetic data by defining a vector space based of model performances, and use a nearest neighbor approach on the bases of correlation to identify the ground truth model. The results show that some similar models that represent almost the same concept show pretty much good correlation but for two different synthetic dataset with the same underlying model there is a high correlation. For those that have completely different model it is low.

Considering this approach and "best performer" as two classification approaches for model fitting, results show that the approach is more accurate than the "best performer" approach, but only for some ground truth models.

Also we discuss the stability of the model performance vector in the space and its uniqueness for different data generation parameters such as sample size, average success rate, number of skills, number of items, examinee and item variance. Results of this experiment show that the performance vector is sensitive to some data generation parameters. Some parameters like number of skills has a tangible effect on model performance pattern unlike some others such as sample size. Generally the pattern slightly changes through predictive performances.

Contents

DEDICATION . . . iii

ACKNOWLEDGEMENTS . . . iv

RÉSUMÉ . . . v

ABSTRACT . . . vi

Contents . . . vii

LIST OF TABLES . . . x

LIST OF FIGURES . . . xi

LIST OF ABBREVIATIONS . . . xii

CHAPTER 1 INTRODUCTION . . . 1
- 1.1 Problem Definition and Challenges . . . 1
  - 1.1.1 Model selection and goodness of fit . . . 2
- 1.2 Research Questions . . . 3
- 1.3 General Objectives . . . 3
- 1.4 Hypotheses . . . 3
- 1.5 Main Contributions . . . 4
- 1.6 Publications . . . 5
- 1.7 Organization Of the Thesis . . . 6

CHAPTER 2 STUDENT MODELLING METHODS . . . 7
- 2.1 ITS and EDM . . . 7
- 2.2 Definitions and concepts . . . 7
  - 2.2.1 Test outcome data, Q-matrix and Skill mastery matrices . . . 7
  - 2.2.2 Partial Order Knowledge Structure(POKS) . . . 8
- 2.3 Skills assessment and item outcome prediction techniques . . . 10
- 2.4 Multi-skills techniques . . . 10
  - 2.4.1 Matrix Factorization . . . 10
  - 2.4.2 Recommender systems and matrix factorization . . . 11

2.4.3 Similarity with recommender systems and assumptions 12
2.4.4 Non-Negative Matrix Factorization(Non-negative Matrix Factorization (NMF)) 12
2.4.5 Types of Q-matrix 14
2.4.6 Deterministic Input Noisy And/Or (DIAN/DINO) 15
2.4.7 Alternate Least-square Factorization (ALS) 15
2.4.8 Other Techniques to Derive Q-matrices from Data 16
2.5 Single skill approaches [should probably be earlier since it is easier to explain] 17
2.5.1 IRT 17
2.5.2 Baseline expected value 17
2.6 Zero skill technique 18
2.6.1 Knowledge Spaces and Partial Order Knowledge Structures (POKS) 18
2.7 Recent improvements 18
2.8 NMF on single skill and multi-skill conjunctive Q-matrix 20
2.8.1 Simulation methodology 21
2.8.2 Results 21
2.8.3 Discussion 23
2.9 Finding the number of latent skills 24
2.9.1 SVD-Based method 24
2.9.2 Wrapper-Based method 25
2.9.3 Results of SVD-Based method 26
2.9.4 Results of Wrapper-Based method 27
2.9.5 Discussion 27
2.10 The refinement of a Q-matrix 29
2.10.1 Q-matrices validation techniques 29
2.10.2 Methodology and data sets 30
2.10.3 Results 32
2.10.4 Recovery rates by the number perturbation 32
2.10.5 Recovery rates by data set 35
2.10.6 Discussion 35
2.11 Improving matrix factorization techniques of student test data with partial order constraints 36

CHAPTER 3 MODEL FITTING AND SIMULATED DATA 39
3.1 Model Fitting 39
3.1.1 Approaches 39
3.1.2 Measures 39

3.1.3 Dynamic vs. static data 40
3.2 On the faithfulness of simulated student performance data 41
3.2.1 Simulated data models 42
3.2.2 methodology 42
3.2.3 Real Datasets 42
3.2.4 discussion 42
3.3 simulated data to reveal the proximity of a model to reality 43
3.3.1 Methodology 44

CHAPTER 4 SYNTHETIC DATA GENERATION 45
4.1 POKS 45
4.2 IRT 46
4.3 Linear NMF Conjunctive 47
4.4 Linear NMF Additive 48
4.5 DINA/DINO 49

CHAPTER 5 Experimental Results 51
5.1 Model fit in a vector space framework 51
5.2 Skills Models 52
5.3 Methodology 54
5.4 Experiment 1: Performance comparison 54
5.4.1 Data sets 56
5.4.2 Predictive performance results 58
5.5 Experiment 2: Sensitivity of the Model performance over data generation parameters 60
5.5.1 Degree of similarity 63

CHAPTER 6 Conclusion and future work 68
6.1 Conclusion 68
6.2 Future Work 68

LIST OF TABLES

Table 2.1 Quantitative comparison between original **Q** matrix and NMF derived matrices $\hat{\mathbf{Q}}$. Results are based on means and standard deviation over 10 simulation runs. . . . 23

Table 2.2 Data sets . . . 31

Table 2.3 Results by individual data set at 1 and 4 perturbations . . . 34

Table 2.4 Results of running NMF, POKS and combined methods on Linear and Bayesian generated datasets . . . 37

Table 5.1 Vector space of accuracy performances . . . 51

Table 5.2 Parameters of the simulation framework . . . 55

Table 5.3 Parameters of the predictive performance framework . . . 55

Table 5.4 Datasets . . . 57

Table 5.5 Degree of similarity between six synthetic datasets based on the correlation 65

Table 5.6 Degree of similarity between six synthetic datasets and the ground truth based on the correlation . . . 66

Table 5.7 Confusion matrix for classification of 210 synthetic datasets on 7 models with Best performer Vs. Nearest neighbor methods . . . 66

Table 5.8 Accuracy of best performer and nearest neighbor classification methods . . 67

LIST OF FIGURES

Figure 2.1 Partial Order Structure of 4 items 9
Figure 2.2 Oriented incidence matrix and Adjacency matrix 9
Figure 2.3 An example for Conjunctive model of Q-matrix 14
Figure 2.4 Items and Q-matrix 19
Figure 2.5 Visual representations of the original **Q** matrix and NMF derived matrices $\hat{\mathbf{Q}}$ 22
Figure 2.6 Singular values of simulated data for a 21 items test. A vertical dashed line at singular value 6 corresponds to the number of underlying latent skill factors. 28
Figure 2.7 Precision of student results predictions from estimated skill matrix (equation 2.10). Error bars are the standard error of the accuracy curves. Experiment is done with simulated data with 6 skills and slip and guess values of 0.1 and 0.2 respectively. 28
Figure 2.8 Average recovery rate by number of perturbations (real and synthetic data). 33
Figure 4.1 Q-matrix and an example of simulated data with this matrix. pale cells represent 1's and red ones represent 0's. 48
Figure 4.2 Additive model of Q-matrix and Corresponding synthetic data 49
Figure 5.1 Skills assessment methods 53
Figure 5.2 Data breakdown of cross validation process 54
Figure 5.3 Item outcome prediction accuracy results of **synthetic data sets** 59
Figure 5.4 Item outcome prediction accuracy results of **real data sets** 59
Figure 5.5 Variation of **Sample Size** Over synthetic data sets 61
Figure 5.6 Variation of **Number of items** Over synthetic data sets 62
Figure 5.7 Variation of **Number of skills** Over synthetic data sets 62
Figure 5.8 Variation of **Item Variance** Over synthetic data sets 64
Figure 5.9 Variation of **Student variance** Over synthetic data sets 64
Figure 5.10 Variation of **Success Rate** Over synthetic data sets 65

LIST OF ABBREVIATIONS

NMF	Non-negative Matrix Factorization
POKS	Partial Order Knowledge Structure
IRT	Item Response Theory
DINA	Deterministic Input Noisy And
DINO	Deterministic Input Noisy Or
SVD	Singular Value Decomposition
ALS	Alternate Least-Square Factorization
E-M	Expectation–Maximization
MCMC	Markov chain Monte Carlo

CHAPTER 1
INTRODUCTION

1.1 Problem Definition and Challenges

Data analysts that wish to build a classification or regression model over new and unknown data are faced with a very wide span of choices. Machine learning techniques nowadays offer the possibility to learn and train a large and an ever growing variety of models from data. Learning techniques such as the E-M algorithm and MCMC methods have contributed to this expansion of models we can learn from data. They allow model parameters estimation that would otherwise represent an intractable problem using standard analytical or optimization techniques.

Along with this increased display of models that can be defined and trained from data, comes the question of deciding which are the most representative of the underlying ground truth. This question is of interest from two perspectives. One is the theoretical and explanatory value of uncovering a model that accounts for observed data. The other perspective is the assumption that the "true" underlying model will better generalize to samples other than the training data. This assumption is commonly supported in physics where some models have a window in the parameter space where they correctly account for observations, and break down outside that window; Newtownian and modern physics are prototypical examples supporting this assumption.

In the machine learning field, the case for the support of the assumption that the closer to the ground truth a model is, the better it will generalize outside the parameter space, is not as evident as it can be in physics. But we do find analogous examples such as the Naïve Bayes classifier under a 0-1 loss function tend to perform very well in spite of the unrealistic assumption of the naïve independence assumption at the root of the approach's name (**?**).

Given that in machine learning, we are often more interested in the predictive power of models than we are in their theoretical and explanatory value, the standard practice is to choose the model with the best predictive performance. And without good theoretical understanding of the domain, we simply hope that it will generalize outside the space covered by our training sample.

This thesis aims to provide a means to assess the fit of the model to the underlying ground truth using a methodology based on synthetic data, and to verify if the approach is better able to identify a model that will generalize outside the parameter space of the training sample. The study is circumscribed to the domain of Educational Data Mining where we find numerous competing models of student skills mastery.

1.1.1 Model selection and goodness of fit

Model selection is the task of selecting a statistical model for a given data from a set of candidate models. Both data and candidate models involve in the design of an experiment which analyses the given data statically. Even selecting candidate models should be done properly where a model that potentially can represent the dataset within its parameters should be considered as a candidate. Given candidate models of similar predictive or explanatory power, the simplest model is most likely to be the best choice. The complexity of the model can be presumed as the number of parameters it contains. The more complex the model is, the better its parameters are learnt to fit the data which also causes over-fitting problem.

On the hand the term "Goodness of fit" for a statistical model describes how well it fits a set of observation. The distance between observed values and the predicted values under the model can be a measure of goodness of fit. The goodness of fit is usually determined using likelihood ratio. There exists different approaches to assess model fit based on the measure of goodness of fit. The consensus is that the model with the best predictive performance is the most likely to be the closest to the ground Truth. Then there are the issues of how sensitive is the model to sample size, noise, and biases that also need to be addressed before we can trust that this model is the best candidate. It can take numerous studies before a true consensus emerges as to which model is the best candidate for a given type of data.

Yet another approach to assess which model is closest to the ground truth is to combine predictive performance analysis of real data with synthetic data. Using synthetic data allows us to validate the sensitivity of the model selection approach to data specific parameters such as sample size and noise. Comparing performance over synthetic and real data has been used extensively to validate models, but we further elaborate on the standard principle of comparison over both types of data by contrasting the predictive performance across types of synthetic data. The hypothesis we make is that the relative performance of different models will be stable by the characteristic of a given type of data, as defined by the underlying ground truth for real data, or by the model that generates the synthetic data. We explore this hypothesis in the domain of Educational Data Mining and the assessment of student skills, where a set of latent skills are mapped to question items and students skill mastery is inferred from item outcome results from test data.

This chapter introduces and defines these concepts, as well as outlines the objectives and main scientific hypotheses of the proposed research. The final section presents the organization of the remainder of this research.

1.2 Research Questions

The following questions are addressed in this thesis:

1. What is the relative performance of student skills assessment models over real and over synthetic data created using the same models?
2. Is the relative performance unique to each synthetic data type (data from the same ground truth model)?
3. Can the relative performance be used to define a method to reliably identify the ground truth behind the synthetic data?
4. How does the method compare with the standard practice of using the model with the best performance? In particular, does the ground truth model identified better generalize over a space of parameter values?

1.3 General Objectives

The general objective of this thesis is to assess model fit on the bases of the goodness of fit measure with predictive performance analysis of real and synthetic data. Fundamentally it can be divided in three sub-objectives: The first objective is to obtain the predictive performance of student skills assessment models over a dataset using the same models. This will create a vector of performances in the performance space. The second one is to assess model fit using the relative performance vector of the synthetic and real data. The third objective is to test the uniqueness and sensitivity of the performance vector on the different data specific conditions such as sample size, nose, average success rate.

1.4 Hypotheses

The research in this thesis tests the following hypotheses:

Hypothesis 1: The relative performances of two datasets with the same underlying models have high correlation.

Hypothesis 2: The best performer model in the predictive performance vector is not necessarily the ground truth.

Hypothesis 3: Datasets with the same model parameters and data specific parameters create unique performance vector. [or Relative performance of each synthetic data type is unique] [or the relative performance of different models will be stable by the characteristic of a given type of data]

Hypothesis 4: Datasets with the same underlying models but different data specific parameters can have different performance vectors.

1.5 Main Contributions

The main contribution of this thesis is assessing model fit using the relative predictive performance vector of synthetic and real data. This method can be applied to different fields of studies but in this research we focused on student test result and few skills assessment models that have emerged mostly in EDM and ITS. The predictive performance of each model is assessed by designing an experiment which learns the model parameters and observes a set of items for a student to predict the rest of items test results of that student. The mean predictive accuracy will be the predictive performance measure. Previous researches compared their predictive performance on a pairwise basis, but few studies have taken a comprehensive approach to compare them on a common basis. In this research we used seven skills assessment models to obtain the predictive performance vector using the same models. This vector can be considered from two perspectives: The first one is a kind of "signature" for a specific data which considers the vector in a two dimensional space which are performances over skills assessment models. The second perspective is a performance vector in the performance space where we have the same number of dimensions as the number of models in the performance vector. They are sharing the same concepts but different presentations.

The next step is to use this performance vector to assess model fit for a real dataset. The standard practice is to pick the "best performer" as the ground truth model. The actual best fitting model may have been overlooked due to an unfortunate choice of the algorithm's parameters. Therefore, the best performer may not be the model that is most representative of the ground truth, but instead it may be the result of contextual factors that make this model outperform the ground truth one. We investigate the question of assessing different model fits using synthetic data by defining a vector space based on model performances, and use a nearest neighbor approach on the bases of correlation to identify the ground truth model. Comparing the performance of synthetic dataset with an specific underlying model and the performance of a real dataset with the same underlying model should show a high correlation.

Still the question of sensitivity of contextual factors should be considered in the comparison of the performance vectors. The other contribution is to test the stability of the "signature" of synthetic datasets with different data specific parameters (such as sample size, average success rate and ect.)

but the same underlying models. Based on the results of this experiment, assessing model fit should be in a condition where both synthetic and real data have the same data specific parameters.

In this study, we used a methodology to assess the model fit. The methodology requires a set of synthetic datasets with each skills assessment models that uses the contextual parameters of real data and a cross validation process to get the predictive performance from both synthetic and real data using the same models. The nearest neighbor of the synthetic datasets performance vectors to the real data performance vector in the performance space identifies the ground truth. This contribution is discussed in detail in Chapter 5.

Considering the proposed (“signature”)method and “best performer” approach as two classification approaches to assess model fit, results show that the “signature” approach is more accurate than the “best performer” , but only for some ground truth models. [not sure to put this or not?]

1.6 Publications

1. **B. Beheshti**, M.C. Desmarais, “Assessing Model Fit With Synthetic vs. Real Data" , Journal Submitted to **Journal of Educational Data Mining**.
2. **B. Beheshti**, M.C. Desmarais, “Goodness of Fit of Skills Assessment Approaches: Insights from Patterns of Real vs. Synthetic Data Sets", Short Paper in **International Educational Data Mining 2015** June 2015, Madrid, Spain, pp: 368-371. [page numbers for all pub.]
3. **B. Beheshti**, M.C. Desmarais, R. Naceur, “Methods to Find the Number of Latent Skills”, short paper in **International Educational Data Mining 2012** July 2012, Crete,Greece. , pp: 81-86.
4. **B. Beheshti**, M.C. Desmarais, “Improving matrix factorization techniques of student test data with partial order constraints", Doctoral consortium in **User Modeling, Adaptation, and Personalization 2012** Aug 2012, Montreal,Canada. , pp: 346-350.
5. M.C. Desmarais, **B. Beheshti**, P. Xu, “The refinement of a q-matrix: assessing methods to validate tasks to skills mapping", Short paper in **International Educational Data Mining 2014** June 2014, London,United Kingdom., pp: 308-3011.
6. M.C. Desmarais, **B. Beheshti**, R. Naceur, “Item to skills mapping: deriving a conjunctive q-matrix from data”, short paper in **Intelligent Tutoring Systems 2012** July 2012, Crete,Greece. , pp: 454-463.
7. M.C. Desmarais, P. Xu, **B. Beheshti**, “Combining techniques to refine item to skills Q-matrices with a partition tree", Full Paper in **International Educational Data Mining 2015** June 2015, Madrid, Spain., pp: 29-36.

8. M.C. Desmarais, R. Naceur, **B. Beheshti**, “Linear models of student skills for static data", Workshop in **User Modeling, Adaptation, and Personalization 2012** July 2012, Montreal, Canada.

1.7 Organization Of the Thesis

We review some of related literature on fundamental concepts in educational data mining and some machine learning techniques that have been used in our experiments in Chapter 2. Chapter 3 discusses about some recent related works about model selection. As a complementary part of the main contribution we explain synthetic data generation approaches in chapter 4. The main contribution of the research is explained in details in Chapter 5 as summarized above. Finally, we conclude and outline future work in Chapter 6.

CHAPTER 2
STUDENT MODELLING METHODS

2.1 ITS and EDM

Educational Data Mining (EDM) deals with the development of methods and techniques to analyze data in an educational context. EDM uses statistical, computational, machine-learning and data mining algorithms to explore different types of educational data for resolving research questions about how students learn. Using educational softwares and also internet based education which is known as e-learning created a big data repository that provides the environment for mining educational data. Therefore using raw data coming from educational systems and applying computational approaches can benefit learners and learning process.

Recent researches in EDM could be put in the following three categories (**?**) :

- Offline education that tries to apply statistical techniques on student's behavior and performance.
- E-learning where web mining techniques are used to improve communication, collaboration, administration, and reporting tools.
- Intelligent tutoring systems (ITS) that tries to adapt teaching to the needs of each particular student.

These research areas are contextually different and they require different types of data, models and techniques depending on the educational environment.

2.2 Definitions and concepts

2.2.1 Test outcome data, Q-matrix and Skill mastery matrices

The student test outcome data can consists in results from exams or from exercises, in the context of an e-learning environment or in paper and pencil form. We use the term *item* to represent exercises, questions, or any task where the student has to apply a skilled performance to accomplish. Student answers are evaluated and categorized as success (1) or failure (0). The data represents a snapshot of the mastery of a student for a given subject matter, as we assume that the student's knowledge state has not changed from the time of anwser to the first question item to the last one.

Test data is defined as an $m \times n$ matrix, $\mathbf{R}$. It is composed of m row *items* and n column students. Note that . If a student successfully answers an item, the corresponding value in the results matrix is 1, otherwise it is 0.

As explained before, this results matrix **R** will be factorized into two other matrices, the skills mastery matrix and the Q-matrix. [Was it explained before? This is hard to understand at this point.]

The skills mastery matrix represents student skills mastery profiles. In this matrix rows are skills and columns represent examinees. A cell with the value of 1 in $\mathbf{S}_{ij}$ indicates that examinee j is mastered with skill i and a value of 0 shows that he does not have the related skill.

The assignment of Skills to Items is described by a Q-matrix that describes which Skill are required by each Item. In a Q-matrix rows represent Items and columns represent Latent factor (Skills). In this matrix, a cell with the value of 1 indicates the Item uses the Skill, while a 0 indicates the Skill is not involved in the item. Barnes, Bitzer, & Vouk in (**?**) were among the early researchers to propose algorithms for automatically discovering a Q-Matrix from data.

2.2.2 Partial Order Knowledge Structure(POKS)

Skills [later on POKS is mentioned as a skill-less technique] are often learnt in a given order. Children learn addition, then subtraction, then multiplication, and so on. That means there should be an order for both skills and items. Based on the probabilistic models, it is possible to build such a structure form a result matrix. For example in figure 5.4 four items are shown in a partial order of knowledge structure. It is required for an examinee to have skills for i_4 in order to solve i_3 and i_2. Also for solving i_1, one should have the required skills for i_2, i_3 and i_4.

This is reflected in the results matrix **R** by closure constraints. Defining a student knowledge state as a subset of all items (i.e. a column vector in **R**), then the space of valid knowledge states is closed under union and intersection according to the theory of Knowledge spaces (**?**). In our study, we will relax this constraint to a closure under union, meaning that the union of any two individual knowledge states is also a valid knowledge state. This means that the constraints can be expressed as a partial order of implications among items (**?**), termed a Partial Order Knowledge Structure (POKS). A few algorithms have been defined to derive such structures from the data in **R** (**??**) and the general idea of the thesis is to use this information to guide factorization algorithms.

A knowledge structure can be represented by an Oriented incidence matrix, O, or by an Adjacency matrix, A. In the oriented incidence matrix, rows are edges and columns are nodes of the graph. The value of -1 shows the start node of an edge and 1 indicates the end of an edge. Therefore for each row(edge) there is only one pair of (-1,1) and the rest of cells are 0. In adjacency matrix both rows and columns are Items and if there is a link between a pair of items(for example $i \rightarrow j$) there should be a 1 in A_{ij} otherwise it is 0. Figure 2.2 shows the corresponded oriented incidence matrix and adjacency matrix of the structure in figure5.4.

$$\begin{tikzcd}
& i_1 : \frac{4}{\frac{12}{3}} + \frac{3}{5} = \frac{8}{5} \arrow[dl, "a"] \arrow[dr, "b"] & \\
i_2 : \frac{4}{\frac{12}{3}} = \frac{4\times 3}{12} = \frac{12}{12} = 1 \arrow[dr, "c"] & & i_3 : 1 + \frac{3}{5} = \frac{8}{5} \arrow[dl, "d"] \\
& i_4 : 2\times\frac{1}{2} = 1 &
\end{tikzcd}$$

Figure 2.1 Partial Order Structure of 4 items

$$\begin{array}{cc}
 & \begin{array}{cccc} i_1 & i_2 & i_3 & i_4 \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}
\end{array}
\qquad
\begin{array}{cc}
 & \begin{array}{cccc} i_1 & i_2 & i_3 & i_4 \end{array} \\
\begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}
\end{array}$$

$$O \qquad\qquad A$$

Figure 2.2 Oriented incidence matrix and Adjacency matrix

2.3 Skills assessment and item outcome prediction techniques

The skills assessment model we compare can be grouped into four categories: (1) the single skill Item Response Theory (IRT) approach, (2) the Knowledge Space frameworks which models a knowledge state as a set of observable items without explicit reference to skills, (3) the matrix factorization approach which decomposes the student results matrix into a Q-matrix that maps items to skills, and a skills matrix that maps skill to students, and which relies on standard matrix algebra for parameter estimation and item outcome prediction, and finally (4) the multi-skills family of DINA/DINO approaches which also refers to a Q-matrix, but incorporates slip and guess factors and relies on different parameter estimation techniques than the matrix factorization method.

Considering these techniques more generally, we can put item outcome prediction techniques that we used in this thesis in the following categories:

— Multi-Skills techniques that rely with Q-matrices to complete their prediction which are also known as skills assessment techniques. Linear techniques used in this research are Deterministic Input Noisy And/Or(DINA/DINO), NMF Conjunctive and NMF additive

— Single skill approaches that consider the student's ability for answering a question and they are not directly deal with a Q-matrix or set of skills. In this research we used Item Response Theory (IRT) and expected as two techniques for item outcome prediction where they consider single skill for each student.

— Zero skill technique that predict item outcome based on observed items. POKS is the technique that is used as a zero skill student model

Meanwhile we defined a trivial approach as a baseline for our evaluation of the results that is called Expected Prediction approach.

However, the models reviewed here assume a static student skills state, as opposed to Knowledge Tracing model and its derivatives **?**, for example.

The details of the specific are described below.

2.4 Multi-skills techniques

2.4.1 Matrix Factorization

Generally, matrix factorization is a method to decompose a matrix into two or more matrices. SVD and NMF are well known examples of such methods. In this research, we focus on means to improve the NMF method.

Assume **R** is a result matrix containing student test result of n items(questions or tests) and m students. NMF decompose the non-negative **R**, as the product of two non-negative matrices as shown in equation(2.1):

$$\mathbf{R} \approx \mathbf{QS} \tag{2.1}$$

where **Q** and **S** are $n \times k$ and $k \times m$ respectively. **Q** is the same as Q-matrix which maps items to skills and **S** represent the skill mastery matrix that represents the mastered skills for each student. k is called as the rank of factorization which is the same as number of latent skills. In some context there is a constraint for the number of latent skills which is : $k < nm/(n+m)$ (**?**)

For example in the following equation , assume that we know the skills behind each item which means we know the exact Q-matrix and also we know the skills mastery as well. In this example the product of **Q** and **S** will reproduces the result matrix. Given a result matrix, we want to decompose this result matrix into the expected Q-matrix and skill mastery matrices.

$$\underset{R}{\begin{array}{cc} & \text{Examinee} \\ \text{Items} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{array}} = \underset{Q}{\begin{array}{cc} & \text{Skills} \\ \text{Items} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{array}} \times \underset{S}{\begin{array}{cc} & \text{Examinees} \\ \text{Skills} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array}}$$

Many algorithms for matrix factorization search the space of solutions to equation (2.1) by gradient descent. These algorithms can be interpreted as rescaled gradient descent, where the rescaling factor is optimally chosen to ensure convergence. Most of factorization algorithms operate iteratively in order to find the optimal factors. At each iteration of these algorithms, the new value of **Q** or **S** (for NMF) is found by multiplying the current value by some factor that depends on the quality of the approximation in Eq. (2.1). It was proved that repeated iteration of the update rules is guaranteed to converge to a locally optimal factorization (**?**).

Besides its use for student skills assessment and for deriving a Q-matrix, matrix factorization is also a widely used technique in recommender systems where we find efforts to improve it. (**?**) shows a brief description of some of these improvements.

2.4.2 Recommender systems and matrix factorization

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

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

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

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

2.4.3 Similarity with recommender systems and assumptions

Equation (2.11) is analoguous to the decomposition of the (*item* $\times$ *user*) votes matrix into two smaller matrices: the (*item* $\times$ *preference*) and (*preference* $\times$ *user*) matrices. However, the the votes matrix is typically sparse and different means have been developed to accomodate matrix factorization techniques with sparse matrices, and to compensate for predictions based on highly uneven number of votes in columns and rows that tend to negatively affect the predictive performance of algorithms.

2.4.4 Non-Negative Matrix Factorization(NMF)

NMF is a factorization technique that decompose a matrix into two matrices. The prominent characteristic of NMF is the non-negative constraint on the decomposed elements. NMF imposes this constraint and consequently all those values in the decomposed elements are non-negative.

The clear point of this decomposition is that there can be different solutions. Although the constraint of non-negative elements eliminate some solutions but still there may be different solutions for this factorization

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

Gradient decent is one of the best known approaches for implementing NMF. If k is less than the minimum of m and n, finding the exact $\mathbf{Q}$ and $\mathbf{S}$ matrices which satisfy $\mathbf{R} = \mathbf{QS}$ can entail a loss of information. Therefore this algorithm tries to get the best estimation for $\mathbf{Q}$ and $\mathbf{S}$ to make $\mathbf{R} \approx \mathbf{QS}$ more accurate. Based on the definition of gradient descent method, a cost function should be defined to quantify the quality of the approximation. This cost function can be a measure of distance between two non-negative matrices $\mathbf{R}$ and $\mathbf{QS}$. It can be the Euclidean distance between these two matrices as shown in equation (2.2) where $\mathbf{Q}_i$ is a row vector of $\mathbf{Q}$ and $\mathbf{S}_j$ is a column vector of $\mathbf{S}$ and $\mathbf{R}_{ij}$ is cell (i,j) of $\mathbf{R}$.

$$||\mathbf{R}-\mathbf{QS}||^2 = \sum_{ij}(\mathbf{R}_{ij} - \mathbf{Q}_i\mathbf{S}_j)^2 \tag{2.2}$$

Another cost function is based on the Kullback-Leibler divergence, which measures the divergence between $\mathbf{R}$ and $\mathbf{QS}$ as shown in equation (2.3).

$$D(\mathbf{R}||\mathbf{QS}) = \sum_{ij}(\mathbf{R}_{ij} log \frac{\mathbf{R}_{ij}}{\mathbf{Q}_i\mathbf{S}_j} - \mathbf{R}_{ij} + \mathbf{Q}_i\mathbf{S}_j) \tag{2.3}$$

In both approaches, the goal is to minimize the cost function where they are lower bounded by zero and it happens only if $\mathbf{R} = \mathbf{QS}$ (**?**). For simplicity we just consider the cost function based on the Euclidean distance.

The gradient descent algorithm used to minimize the error is iterative and in each iteration we expect a new estimation of the factorization. We will refer to the estimated Q-matrix as $\hat{\mathbf{Q}}$ and the estimated Skill mastery matrix as $\hat{\mathbf{S}}$. The iterative gradient descent algorithm should change $\mathbf{Q}$ and $\mathbf{S}$ to minimize the cost function. This change should be done by an update rule. Lee and Seung (**?**) found the following update rule in equation (2.4). These update rules in equation (2.4) guarantee that the Euclidean distance $||\mathbf{R}-\mathbf{QS}||$ is non increasing during the iteration of the algorithm.

$$\hat{\mathbf{S}} \leftarrow \hat{\mathbf{S}}\frac{(\hat{\mathbf{Q}}^T\mathbf{R})}{(\hat{\mathbf{Q}}^T\hat{\mathbf{Q}}\hat{\mathbf{S}})} \qquad \hat{\mathbf{Q}} \leftarrow \hat{\mathbf{Q}}\frac{(\mathbf{R}\hat{\mathbf{S}}^T)}{(\hat{\mathbf{Q}}\hat{\mathbf{S}}\hat{\mathbf{S}}^T)} \tag{2.4}$$

The initial value for Q and S are usually random but they can be adjusted to a specific method of

NMF library to find the best seeding point.

2.4.5 Types of Q-matrix

There are three models for the Q-matrix which are useful based on the context of the problem domain. The most important one is the conjunctive model of the Q-matrix which is the standard interpretation of the Q-matrix. In figure 5.3 an example of conjunctive model of Q-matrix is shown. Examinee e_1 answered item i_1 and item i_4 because he has mastered in the required skills but although he has skill s_1 he couldn't answer item i_3 which requires skill s_2 as well.

Barnes (**?**) proposed equation 2.5 for this model where the operator $\neg$ is the boolean negation that maps 0 values to 1 and other values to 0. In this way, if an examinee that mastered all required skills for an item, he will get 1 in the result matrix otherwise he will get a 0 value, even if the required skills are partially mastered.

In fact if we apply a boolean negation function to both sides of the equation 2.5, we will see that the $\neg\mathbf{R}$ matrix is a product of two matrices, $\mathbf{Q}$ and $\neg\mathbf{S}$

$$\mathbf{R} = \neg\left(\mathbf{Q}\left(\neg\mathbf{S}\right)\right) \tag{2.5}$$

In this research we focus on the conjunctive model of Q-matrix but there are two other types of Q-matrices: disjunctive and compensatory. In the disjunctive model, there is a disjunction between skills of an item. At least one of the required skills should be mastered in order to have a success in answering the related item.

Compensatory or additive model of skills is an interpretation of a Q-matrix where skills have weights to yield a success for that item. For example, considering an item requires two skills a and b with the same weight each. Then each skill will contribute equally to yield a success of the item. In the compensatory model of Q-matrix, each skills increase the chance of success based on

$$\text{Items}\quad \begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \end{array}\overset{\text{Examinee}}{\begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \\ \hline \end{array}}\! \begin{array}{c} \\ \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right] \end{array}$$

$$R$$

$$\text{Items}\quad \begin{array}{c|ccc} & \multicolumn{3}{c}{\text{Skills}} \\ & s_1 & s_2 & s_3 \\ i_1 & 1 & 0 & 0 \\ i_2 & 0 & 1 & 1 \\ i_3 & 1 & 1 & 0 \\ i_4 & 1 & 0 & 1 \end{array}$$

$$Q$$

$$\text{Skills}\quad \begin{array}{c|cccc} & \multicolumn{4}{c}{\text{Examinees}} \\ & e_1 & e_2 & e_3 & e_4 \\ s_1 & 1 & 0 & 1 & 0 \\ s_2 & 0 & 1 & 1 & 1 \\ s_3 & 1 & 1 & 0 & 0 \end{array}$$

$$S$$

Figure 2.3 An example for Conjunctive model of Q-matrix its weight.

2.4.6 Deterministic Input Noisy And/Or (DIAN/DINO)

[This one differs because it assumes an existing Q-matrix.]

The other skills assessment model we consider are based on what is referred to as Deterministic Input Noisy And/Or (DINO/DINA) **?**. They also rely on a Q-matrix. The DINA model (Deterministic Input Noisy And) corresponds to the conjunctive model whereas the DINO (Deterministic Input Noisy Or) corresponds to the disjunctive one, where the mastery of a single skill is sufficient to succeed an item. The acronyms makes reference to the AND/OR gates terminology.

These models predict item outcome based on three parameters: the slip and guess factors of items, and the different "gate" function between the student's ability and the required skills. The gate functions are equivalent to the conjunctive and disjunctive vector product logic described for the matrix factorization above. In the DINA case, if all required skills are mastered, the result is 1, and 0 otherwise. Slip and guess parameters are values that generally vary on a $[0,0.2]$ scale. In the DINO case, mastery of any skills is sufficient to output 1. Assuming ξ is the output of the corresponding DINA or DINO model and s_j and g_j are the slip and guess factors, the probability of a successful outcome to item X_{ij} is:

$$P(X_{ij}=1 \mid \xi_{ij}) = (1-s_j)^{\xi_{ij}} g_j^{1-\xi_{ij}} \tag{2.6}$$

The DINO model is analog to the DINA model, except that mastery follows the disjunctive framework and therefore $\xi_{ij} = 1$ if *any* of the skills required by item j are mastered by student i.

A few methods have been developed to estimate the slip and guess parameters from data and we use the one implemented in the R CDM package (**?**).

2.4.7 Alternate Least-square Factorization (ALS)

The Alternate Least-Square Factorization (ALS) method is defined in **?** which is also a factorization method. Starting with the results matrix $\mathbf{R}$ and an initial Q-matrix, $\mathbf{Q}_0$, a least-squares estimate of the skills matrix $\hat{\mathbf{S}}_0$ can be obtained by:

$$\hat{\mathbf{S}}_0 = (\mathbf{Q}_0^{\mathrm{T}} \mathbf{Q}_0)^{-1} \mathbf{Q}_0^{\mathrm{T}} \mathbf{R}$$

The initial matrix $\mathbf{Q}_0$ is the expert defined Q-matrix. Then, a new estimate of the Q-matrix, $\hat{\mathbf{Q}}_1$, is again obtained by the least-squares estimate:

$$\hat{\mathbf{Q}}_1 = \mathbf{R}\hat{\mathbf{S}}_0^{\mathrm{T}}(\hat{\mathbf{S}}_0\hat{\mathbf{S}}_0^{\mathrm{T}})^{-1}$$

And so on for estimating $\hat{\mathbf{S}}_1$, $\hat{\mathbf{Q}}_2$, ... Alternating between equations (2.4.7) and (2.4.7) yields progressive refinements of the matrices $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{S}}_i$ that more closely approximate $\mathbf{R}$ in equation (2.11). The convergence is based on a predefined delta between iterations. In our experiments we consider delta as 0.001 that makes the algorithm to converge after few iterations .This performance makes the technique many times more efficient than factorizations that rely on gradient descent, for example.

The ALS factorization is *compensatory* in the above description. A *conjunctive* version can be obtained by inverting the values of the $\mathbf{R}$ matrix **?**.

It is worth mentioning that, by starting with non negative matrices $\mathbf{Q}_0$ and $\mathbf{R}$, the convergence process will generally end with positive values for both matrices $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{S}}_i$. The vast majority of values obtained are between 0.5 and 1.5 if both the results matrix and the initial Q-matrix have 0,1 values. Also regularization terms could be used in the implementation of the algorithm to force non-negative or integer values.

Note that $(\mathbf{Q}_0^{\mathrm{T}}\mathbf{Q}_0)$ or $(\hat{\mathbf{S}}_0\hat{\mathbf{S}}_0^{\mathrm{T}})_0$ may not be invertable, for example in the case where the matrix $\mathbf{Q}_0$ is not column full-rank, or the matrix $\mathbf{S}_0$ is not row full-rank. This is resolved by adding a very small Gaussian noise before attempting the inverse. Ensuring the choice of a relatively insignificant noise does not affect the end result for our purpose.

2.4.8 Other Techniques to Derive Q-matrices from Data

[The derivation of a Q-matrix from data is a prerequesite for some models whereas it sometimes is a given for others. This should probably be explained beforehand. It also makes a good argument for the work on the induction of a Q-matrix from data since, without it, performance cannot be assessed for these models.]

Cen et al. (**??**) have used Learning FactorAnalysis technique(LFA) technique to improve the initially hand built Q-matrix which maps fine-grained skills to questions. They used log data which is based on the fact that the knowledge state of student dynamically changes over time as the student learns. In the case of static data of student knowledge, Barnes (**?**) developed a method for this mapping which works based on a measure of the fit of a potential Q-matrix to the data. It was shown to be successful as well as Principle Component Analysis for skill clustering analysis.

In the experiments of Winters et al. (**?**), different standard clustering techniques for finding the Q-matrix were applied on different types of datasets. Some datasets from SAT topics such as

Mathematics, Biology and French, to computer science exams, and to different trivia topics were tested with the clustering techniques. It was shown that those techniques were successful at separating items that belongs to totally different topics, for example mathematics and french. For topics without any skill behind them(essentially they are facts and the only skill required is to memorize them), topic separation was less successful.

2.5 Single skill approaches **[should probably be earlier since it is easier to explain]**

2.5.1 IRT

The **IRT family** is based on a logistic regression framework. It models a single latent skill (although variants exists for modeling multiple skills) **?**. Each item has a difficulty and a discrimination parameter.

IRT assumes the probability of success to an item X_j is a function of a single ability factor θ:

$$P(X_j = 1 \mid \theta) = \frac{1}{1 + e^{-a_j(\theta - b_j)}}$$

In the two parameter form above, referred to as IRT-2pl, parameter a represents the item discrimination and parameter b represents the item difficulty.

The ability of a single student, θ_i is estimated by maximizing the likelihood of the observed response outcomes probabilities:

$$P(X_1, X_2, ..., X_j, \theta_i) = \prod_j P(X_j | \theta_i)$$

This corresponds to the usual logistic regression procedure.

The specific IRT skills assessment version is the Rash model, for which the discrimination parameter a is fixed to 1. Fixing this parameter reduces over fitting, as the discrimination can sometimes take unrealistically large values. Note however that we do generate synthetic data from the more general IRT-2pl model to make this data more realistic.

2.5.2 Baseline expected value

As a baseline model, we use the expected value of success to an item i by student j, as defined by a product of odds:

$$O(X_{ij}) = O(X_i)O(S_j)$$

where $O(X_i)$ are the odds of success to item i and $O(S_j)$ are the odds of success of student j. Both odds can be estimated from a sample. Recall that the transformation of odds to probability is $P(X) = 1/(1+O(X))$, and conversely $O(X) = P(X)/(1-P(X))$. probabilities are estimated using the Laplace correction: $P(X) = (f(x{=}1)+1)/(f(x{=}1)+f(x{=}0)+2)$

2.6 Zero skill technique

[Needs an introduction.]

2.6.1 Knowledge Spaces and Partial Order Knowledge Structures (POKS)

The knowledge spaces approach models a knowledge domain as a set of observable items. An individual's knowledge state is represented as a subset of these items. The "knowledge space" determines the valid subsets of items. There are no explicit latent skills, but skills assessment can be obtained from the estimated item outcomes given a Q-matrix.

The model adopted to assess knowledge is POKS (Partial Order Knowledge Structures), which is a more constrained version of Knowledge Spaces theory **?**. POKS assumes that items are learned in a strict partial order. It uses this order to infer that the success to hard items increases the probability of success to easier ones, or conversely, that the failure to easy items decreases the chances of success to harder ones.

The structure of the partial order of items is obtained from a statistical hypothesis test that reckons the existence of a link between two items, say $A \rightarrow B$, on the basis of two Binomial statistical tests $P(B|A) > \alpha_1$ and $P(\neg A|\neg B) > \alpha_1$ and under a predetermined alpha error ($\alpha < \alpha_2$). The values of $\alpha_1 = .85$ and $\alpha_2 = .10$ are chosen in this study across all experiments.

A student knowledge state is represented as a vector of probabilities, one per item. Probabilities are updated under a Naive Bayes assumption as simple posterior probabilities given observed items.

2.7 Recent improvements

Curriculum design can be a complex task and an expert blind spots in designing curricula is possible. It is desirable to have an alternative to human sequenced curricula. To do so, there should be a model designed for this purpose which maps latent factors (simply referred to as skills, concepts or facts in the field of ITS) to items (simply referred to as questions) (**??**). Figure 2.4 shows an example of this mapping which is named Q-matrix. Figure 2.4(a) shows 4 items and each item requires different skills (or combination of skills) to be successfully answered. Assuming 3 skills such as fraction multiplication (s_1), fraction addition (s_2) and fraction reduction(s_3), these questions can be mapped to skills like the Q-matrix represented in figure 2.4(b). Such a mapping is desirable and very important in student modelling; because optimal order of problems(sequence of repetition and presentation) can be determined by this model since it allows prediction of which item(question or problem) will cause learning of skills that transfer to the other items most efficiently. It can also be used to assess student knowledge of each concept, and to personalize the tutoring process according to what the student knows or does not know. For example, Heffernan et al. in (**?**) have developed an intelligent tutoring system (the ASSISTment system) that relies on fine grain skills mapped to items.

$$
\begin{array}{ll}
i_1 & \frac{4}{\frac{12}{3}} + \frac{3}{5} = \frac{8}{5} \\
i_2 & \frac{4}{\frac{12}{3}} = \frac{4\times 3}{12} = \frac{12}{12} = 1 \\
i_3 & 1 + \frac{3}{5} = \frac{8}{5} \\
i_4 & 2 \times \frac{1}{2} = 1
\end{array}
$$

(a) Items with different skills

$$
\text{Skills} \qquad
\text{Items}\;
\begin{array}{c}
\\ i_1 \\ i_2 \\ i_3 \\ i_4
\end{array}
\begin{array}{c}
\begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\
\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}
\end{array}
$$

(b) Q-matrix

Figure 2.4 Items and Q-matrix

Different techniques and methods in the field of data mining were used to derive a Q-matrix. Matrix factorization is one of the most important one in this area. Matrix factorization is a method to decompose a matrix into two or more matrices. It is an important technique in different fields such as skill modelling, bioinformatics, and vision, to name but a few. It has achieved great results in each of these fields. For skills assessment, Nguyen et al. (**?**) have shown that the approach can lead to assessments that reach prediction accuracies comparable and even better than well established techniques such as Bayesian Knowledge Tracing (**?**). Matrix factorization can also lead to better means for mapping which skills can explain the success to specific items.

It has been shown that this mapping can be derived by Matrix Factorization techniques (**??**) on a test result data with different topic such as history, biology, math and french. Considering a single skill for each topic, the derived Q-matrix can represent a single skill per item mapping. Dealing with Q-matrices that success or failure of an item depends on more than one skill requires more complicated interpretation. A general contribution for such a problem is identifying a optimize number of skills for a set of items. It has been reported in (**?**) that number of skills should conform to : $r < nm/(n+m)$ where r, n and m is the number of skills, items and students respectively. Finding the best number of skills which counts all the essential and useful skills is always desired. The other part of this thesis is to define approaches to find the optimum number of skills for creating a Q-matrix. Finally, the last step for improving the quality of an expert defined Q-matrix is to verify and refine its accuracy based on some approaches that we will discuss in next chapters.

The rest of this chapter presents some researches that are related to the main contribution of this thesis. The general perspective of section 2.8 is to find a way for deriving the Q-matrix from data, along with a Skills mastery matrix. We aim to find a method to derive these matrices for different types of Q-matrices. This should be a valuable and challenging task. Finding the number of latent skills (i.e. the common dimension between matrices **Q** and **S**) is another important task that is described in section 2.8 . Finally, in section 2.10 we will introduce methods to validate tasks to skills mapping which is also applicable for the refinement of a Q-matrix.

The following sections will describe these completed researches.

2.8 NMF on single skill and multi-skill conjunctive Q-matrix

A few studies have shown that a mapping of skills to items can, indeed, be derived from data (**??**). Winters et al. showed that different data mining techniques can extract item topics, one of which is matrix factorization (**?**). They showed that NMF works very well for simulated data, but the technique's performance with real data was degraded. Their study showed that only highly distinct topics such as mathematics and French can yield a perfect mapping. Biology and history were the other topics in this research which were named as general topics. Clustering of these topics was not accurate because they are factual knowledge and they are not comparable with some topics like mathematics.

This study was later corroborated by Desmarais (**?**) who also used simulated data to show that the low discrimination power of some topics might be explained by their lower weight in terms of skill factors, when compared to other factors such as item difficulty and student ability.

The factorization methods in the studies mentioned above rely on the standard matrix operators which was in formula 2.1 and therefore can be considered as compensatory models of Q-matrix which was explained in the previous chapter. Actually, in these study all items rely on one skill and there was not a combination of skills for any item, thus the simple model of factorization will work.

To follow up the on previous studies (**???**) we can apply NMF technique to equation 2.5. The goal of this task is to determine if NMF can successfully derive a conjunctive model of Q-matrix , where skills do not add up to increase the chances of success to an item, but instead are necessary conditions.

2.8.1 Simulation methodology

The assessment of the NMF performance to infer a Q-matrix from simulated test data such as figure 4.1's is conducted by comparing the predefined Q-matrix, $\mathbf{Q}$, as shown in figure 4.1, with the $\hat{\mathbf{Q}}$ matrix obtained in the NMF of equation 2.5.

As mentioned above, the negation operator is applied over the simulated test data and the NMF algorithm is carried over this data. We used the R NMF package (**?**) and the Brunet NMF algorithm.

We defined a specific method for the quantitative comparison of the matrix $\hat{\mathbf{Q}}$ with $\mathbf{Q}$. First, the $\hat{\mathbf{Q}}$ matrix contains numerical values on a continuous scale. To simplify the comparison with matrix $\mathbf{Q}$, which is composed of $\{0, 1\}$ values, we discretize the numerical values of $\hat{\mathbf{Q}}$ by applying a clustering algorithm to each item in $\hat{\mathbf{Q}}$, forcing two clusters, one for 0's and one for 1's. For example, item 1 in the NMF inferred matrix of figure 2.5 (which we explain later) corresponds to a vector of six numerical values, say $\{1.6, 1.7, 0.0015, 0.0022, 0.0022, 0.0018\}$. This vector clearly cluster into the $\{1, 1, 0, 0, 0, 0\}$ vector of item 1 in figure 2.5. The K-means algorithm is used for the clustering process of each item and we use the kmeans routine provided in R (version 2.13.1).

Then, to determine which skill vector (column) of the $\hat{\mathbf{Q}}$ matrix corresponds to the skill vector of the $\mathbf{Q}$ matrix, a correlation matrix is computed and the highest correlation of each column vector $\hat{\mathbf{Q}}$ is in turn matched with the corresponding unmatched column in $\mathbf{Q}$.

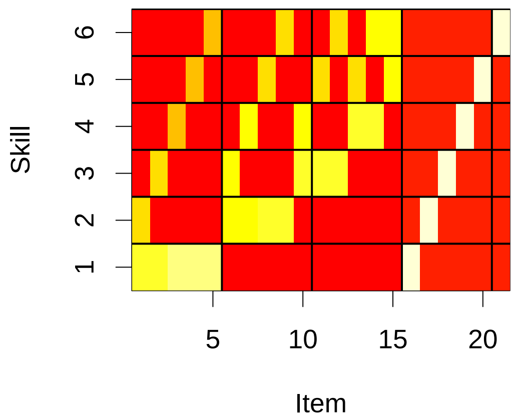
We will use visual representations of the raw and the "discretized" (clustered) $\hat{\mathbf{Q}}$ matrix to provide an intuitive view of the results, as well as a quantitative measures of the fit corresponding to the average of the correlations between the matched skills vectors $\hat{\mathbf{Q}}$ and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

2.8.2 Results

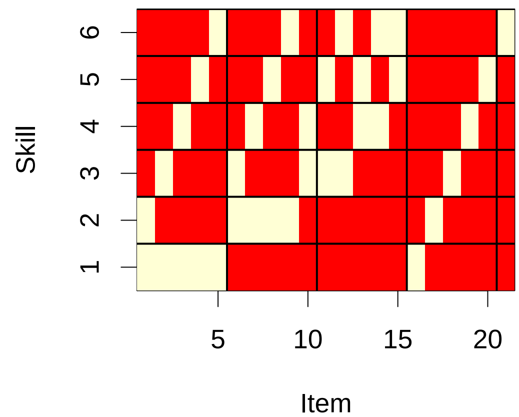
In order for the mean and variance of the simulated data to reflect realistic values of test data, the skill difficulty and examinee ability parameters are adjusted such that the average success rate is close to 60%. Examinee ability is combined with the skill difficulty vectors to create a probability matrix of the same dimensions as $\mathbf{S}$, from which $\mathbf{S}$ is obtained.

Figure 2.5(a) shows a heat map of the matrix $\mathbf{Q}$ inferred from an ideal response pattern of 200 simulated examinees. Skill difficulties were set at (0.17, 0.30, 0.43,0.57, 0.70, 0.83) and examinee mean ability and standard deviation respectively at 0 and 0.5. The discretized version of figure 2.5(a)'s matrix is shown in figure 2.5(b) and it is identical to the underlying matrix $\mathbf{Q}$ in figure 4.1(a).

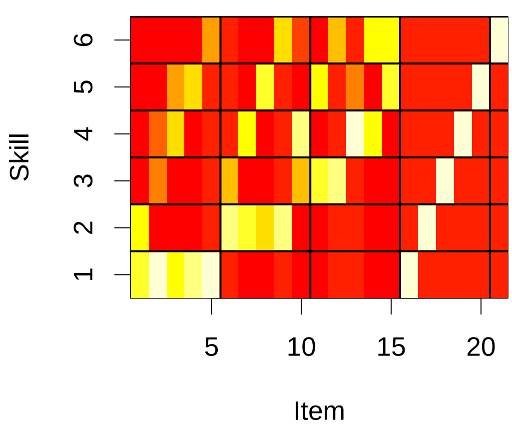
Figure 2.5(a) and 2.5(d) shows the effect of adding slip and guess parameters of 0.2 for each. The mapping to the underlying matrix $\mathbf{Q}$ degrades as expected, but remains relatively accurate.



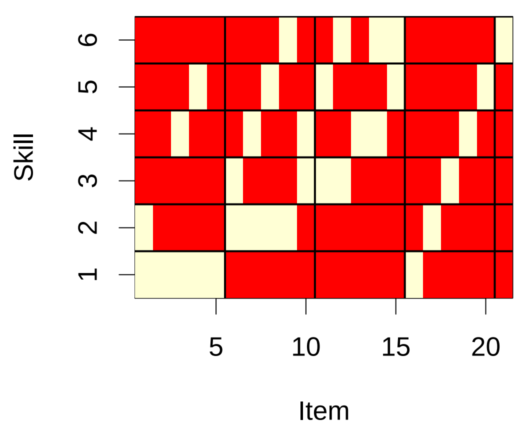
(a) Matrix $\hat{\mathbf{Q}}$ without slip and guess factors ($r = 1$)



(b) Discretized $\hat{\mathbf{Q}}$ without slip and guess factors ($r = 1$)



(c) Matrix $\hat{\mathbf{Q}}$ with slip and guess factors of 0.2 ($r =$ 0.91)



(d) Discretized $\hat{\mathbf{Q}}$ for slip and guess of 0.2 ($r =$ 0.93). Three out of 36 skill requirements are incorrectly mapped in this example.

Figure 2.5 Visual representations of the original $\mathbf{Q}$ matrix and NMF derived matrices $\hat{\mathbf{Q}}$

Table 2.1 reports the results of the quantitative comparison between the $\mathbf{Q}$ matrix and the $\hat{\mathbf{Q}}$ matrix inferred as a function of different slip and guess parameters. These results are based on 10-fold simulations. The mean of the Pearson correlation coefficient (r) between $\mathbf{Q}$ and $\hat{\mathbf{Q}}$ is reported for the discretized version of $\hat{\mathbf{Q}}$ obtained with the clustering algorithm described in previous section. In addition, the error rate as computed by this formula is also provided:

$$Err = \frac{\sum_{ij} |w_{ij} - q_{ij}|}{2 \cdot \sum_{ij} |q_{ij}|} \tag{2.7}$$

Where w_{ij} and q_{ij} are respectively the (i, j) cells of the matrices $\hat{\mathbf{Q}}$ and $\mathbf{Q}$. The error rate will be 0 for a perfectly matched $\mathbf{Q}$ and 1 when no cells match. A value of 0.5 indicates that half of the non-zero cells are correctly matched. For the matrix $\mathbf{Q}$, the estimated random error rate is 71%.

The 0 *slip* and 0 *guess* condition (first line) correspond to figures 2.5(a) and 2.5(b), whereas the corresponding 0.2–0.2 condition (line 3) correspond to figures2.5(c) and 2.5(d).

Up to the 0.2–0.2 slip-guess condition, the skill mapping stays relatively close to perfect. On average, approximately only 2 or 3 skills requirements are wrongly assigned out of the 36 skills requirements (7%) at the 0.2–0.2 condition. However, the error rate increases substantially at the 0.3–0.2 slip-guess condition, and at the 0.4–0.2 condition, the quality of the match is considerably degraded with an average of 13/36 wrong assignements (36%).

2.8.3 Discussion

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

Of course, with a slip factor of 0.3 and a guess factor 0.2, half the values in the results become inconsistent with the Q-matrix. A substantial degradation is therefore not surprising. But in this experiment with simulated data, we have a number of advantages that are lost with real data: the number of skills is known in advance, no item has more than two conjunctive skills, skills are

Table 2.1 Quantitative comparison between original $\mathbf{Q}$ matrix and NMF derived matrices $\hat{\mathbf{Q}}$. Results are based on means and standard deviation over 10 simulation runs.

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

independent, and surely other factors will arise to make real data more complex. Therefore, we can expect that even if real data does not have a 50% rate of inconsistent results with the examinees' skills mastery profile, other factors might make the induction of the Q-matrix subject to errors of this scale.

2.9 Finding the number of latent skills

We do not need to identify all the skills behind an item in order to use the item outcome for assessment purpose. As long as we can establish a minimally strong tie from an item to a skill, this is a sufficient condition to use the item in the assessment of that skill. But knowledge that there is a fixed number of determinant factors to predict item outcome is a useful information. For example, if a few number of skills, say 6, are meant to be assessed by a set of 20 questions items, and we find that the underlying number of determinant latent factors behind these items is very different than 6, then it gives us a hint that our 6-skills model may not be congruent with the assessment result.

In an effort towards the goal of finding the skills behind a set of items, we investigate two techniques to determine the number of dominant latent skills. The SVD is a known technique to find latent factors. The singular values represent direct evidence of the strength of latent factors. Application of SVD to finding the number of latent skills is explored. We introduce a second technique based on a wrap- per approach. Linear models with different number of skills are built, and the one that yields the best prediction accuracy through cross validation is considered the most appropriate. The results show that both techniques are effective in identifying the latent factors over synthetic data. An investigation with real data from the fraction algebra domain is also reported. Both the SVD and wrapper methods yield results that have no simple interpretation.

2.9.1 SVD-Based method

SVD is a well known matrix factorization technique that decomposes any matrix, $\mathbf{A}$, into three sub-matrices:

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}} \tag{2.8}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and their column vectors respectively represent the eigenvectors of $\mathbf{A}\mathbf{A}^{\mathbf{T}}$ and $\mathbf{A}^{\mathbf{T}}\mathbf{A}$. $\mathbf{D}$ is a diagonal matrix that contains the singular values. They are the square root of the eigenvalues of the eigenvectors and are sorted in a descending order. Because the singular values represent scaling factors of the unit eigenvectors in equation 2.8, they are particularly useful in finding latent factors that are dominant in the data. This is demonstrated with simulated data below. The simulated data in this experiment is generated based on the same pattern on the previous section. Again we have all combination of skills for items, at most 2 skills per item and all other factors like slip and guess.

2.9.2 Wrapper-Based method

We introduce a second method to determine the number of dominant skills behind items based on a wrapper approach. In statistical learning, the wrapper approach refers to a general method for selecting the most effective set of variables by measuring the predictive performance of a model with each variables set (see (**?**)). In our context, we assess the predictive performance of linear models embedding different number of latent skills. The model that yields the best predictive performance is deemed to reflect the optimal number of skills.

The wrapper method requires a model that will predict item outcome. A linear model of skills is defined for that purpose on the basis of the equation 2.1. This model represents a compensatory interpretation of skills modelling, where each skill contributes additively to the success of an item. The conjunctive model of skills was presented in 2.5. To estimate the optimal number of skills, the wrapper model can either correspond to equation2.1 or 2.5. We will focus our explanations around equation 2.1, but they obviously apply to 2.5 if $\mathbf{R}$ and $\mathbf{S}$ are negated.

This model states that, given estimates of $\mathbf{Q}$ and $\mathbf{S}$, we can predict $\mathbf{R}$. We refer to these estimates as $\hat{\mathbf{Q}}$ and $\hat{\mathbf{S}}$, and to the predictions as $\hat{\mathbf{R}} = \hat{\mathbf{Q}}\hat{\mathbf{S}}$. The goal is therefore to derive estimates of $\hat{\mathbf{Q}}$ and $\hat{\mathbf{S}}$ with different number of skills and measure the residual difference between $\mathbf{R}$ and $\hat{\mathbf{R}}$.

First, $\hat{\mathbf{Q}}$ is learned from an independent set of training data. Then, $\hat{\mathbf{S}}$ is learned from the test data, and the residuals are computed [c0].

An estimate of $\hat{\mathbf{Q}}$ is obtained through NMF. Details on applying this technique to the problem of deriving a Q-matrix from data is found in (**?**) and we limit our description to the basic principles and issues here.

The NMF technique requires to choose a rank for the decomposition, which corresponds in our situation to the number of skills (i.e. number of columns of $\mathbf{Q}$ and number of rows of $\mathbf{S}$). Because NMF constrains $\mathbf{Q}$ and $\mathbf{S}$ to non-negative values, their respective interpretation as a Q-matrix and a as student skills assessments is much more natural than other matrix factorization techniques such as Principal Component Analysis, for example. However, multiple solutions exists to this

c0. Note that computing $\hat{\mathbf{S}}$ from the test data raises the issue of over-fitting, which would keep the accuracy growing with the number of skills regardless of the "real" number of skills. However, this issue is mitigated by using independent learning data for $\hat{\mathbf{Q}}$, without which, we empirically observed, the results would deceive us: in our experiments using both $\mathbf{Q}$ and $\mathbf{S}$ from NMF while increasing the rank of the factorization (number of skills), ends up increasing prediction accuracy even after we reach beyond the "real" number of skills. This can reasonably be attributed to over-fitting.

factorization and there are many algorithms that can further constrain solutions, namely to force sparse matrices. Our experiment relies on the R package named NMF and the Brunet algorithm (**?**).

Once $\hat{\mathbf{Q}}$ is obtained, then the values of $\hat{\mathbf{S}}$ can be computed through linear regression. Starting with the overdetermined system of linear equations:

$$\mathbf{R} = \hat{\mathbf{Q}}\hat{\mathbf{S}} \tag{2.9}$$

which has the same form as the more familiar $y = \mathbf{X}\beta$ (except that y and β are generally vectors instead of matrices), it follows that the linear least squares estimate is given by:

$$\hat{\mathbf{S}} = \left(\hat{\mathbf{Q}}^T\hat{\mathbf{Q}}\right)^{-1}\hat{\mathbf{Q}}^T\mathbf{R} \tag{2.10}$$

Equation 2.10 represents a linear regression solution which minimizes the residual errors $\left(\left\|\mathbf{R} - \hat{\mathbf{Q}}\hat{\mathbf{S}}\right\|\right)$.

2.9.3 Results of SVD-Based method

The results of the SVD method are shown in figure 2.6. The x is the index of the singular value, and the y axis is its actual value. Recall that the singular values of SVD indicate the strength of latent factors.

Three conditions are reported in figure 2.6. The y values at 1 on the x scale are truncated on the graph to allow a better view of the interesting region of the graph, but the highest value is from the [guess=0, slip=0] condition and the lowest is for the random condition. The random curve condition can be obtained by simulating random $\{0, 1\}$ values and ensuring that the overall average score of the results matrix reflects the original's data average. In this random condition, the slope from singular value 2 to 21 remains relatively constant, suggesting no specific number of skills. In condition [guess=0, slip=0], a sharp drop occurs between singular values of 6 and 7. Then the slope remains relatively constant from values 8 to 21. The largest drop is clearly at value 6 which corresponds to the underlying number of skills. In the third condition [guess=0.2, slip=0.1], the largest drop still remains visible between 6 and 7, but not as sharp as for the noiseless condition, as expected.

In other experiments with various number of skills, not reported here due to space constraints, we observed similar patterns. Another observation is that the random curve intersects with the other two after the number of underlying latent skills (after 6 in figure 2.6's experiment).

Therefore, the SVD method does allow for the identification of the number of skills with synthetic data, at least up to the [guess=0.2, slip=0.1] level.

2.9.4 Results of Wrapper-Based method

We would expect the model with the correct number of skills to perform the best, and models with fewer skills to under- perform because they lack the correct number of latent skills to reflect the response patterns. Models with greater number of skills than required should match the performance of the correct number model, since they have more representative power than needed, but they run higher risk of over- fitting the data and could therefore potentially show lower accuracy in a cross-validation. However, the skills matrix $\hat{\mathbf{S}}$ obtained through equation 2.10 on the test data could also result in over-fitting that will increase accuracy this time.

Figure 2.7 shows the percentage of correct predictions of the models as a function of the number of skills. Given that predictions are {0,1}, the percentage can be computed as $\left(\|\mathbf{R}-\hat{\mathbf{Q}}\hat{\mathbf{S}}\|\right)/mn$, where m and n are the number of rows and columns of $\mathbf{R}$.

The results confirm the conjectures above: the predictive accuracy increases until the underlying number of skills is reached, and it almost stabilizes thereafter. Over-fitting of $\hat{\mathbf{S}}$ with the test data is apparently not substantial. It is interesting to note that the accuracy increments of figure 2.7 are relatively constant between each skill up to 6. This is also what we would expect since every skill in the underlying Q-matrix has an equivalent weight to all others. We expect that differences in increments indicate differences in the weights of the skills. This could either stem from the structure of the Q-matrix (for e.g., more items can depend on one skill than on another), or on the criticality of the skill over its item outcome.

2.9.5 Discussion

This research showed that Both the SVD and the wrapper methods provide strong cues of the number of underlying skills with simulated student test data. However, for the Vomlel data set, both methods yield results that are much more ambiguous. Instead of the 7 skills that were identified by experts over the 17 items set, the SVD method suggests only 2 skills if we rely on the intersection with the random data curve, and no clear number if we look for a change of slope after skill 2. The wrapper method shows data that is also consistent with 2 skills to the extent that a drop of accuracy is observed at 3 skills, but a rise of accuracy up to 8 skill draws an interpretation closer to the experts' 7 skills set.

An important difference between the SVD and the wrapper methods has to do with the independence of skills. For SVD, orthogonality of the singular matrices $\mathbf{U}$ and $\mathbf{V}$ in equation (2.1) forces latent factors to be independent. NMF does not require latent factors to be independent. The or-

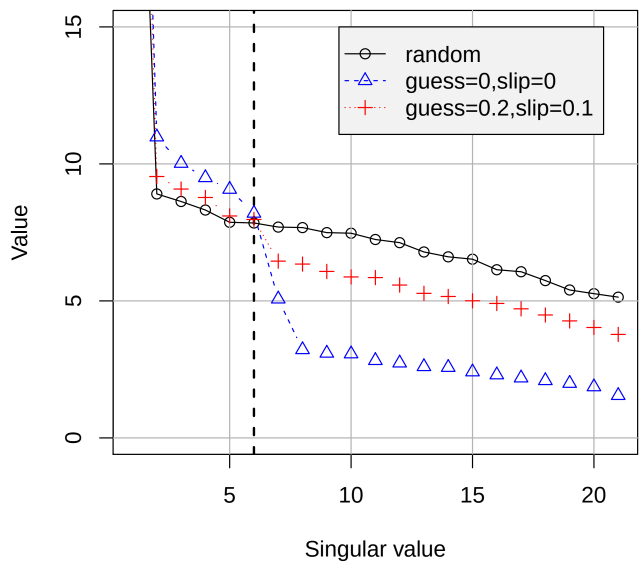


Figure 2.6 Singular values of simulated data for a 21 items test. A vertical dashed line at singular value 6 corresponds to the number of underlying latent skill factors.

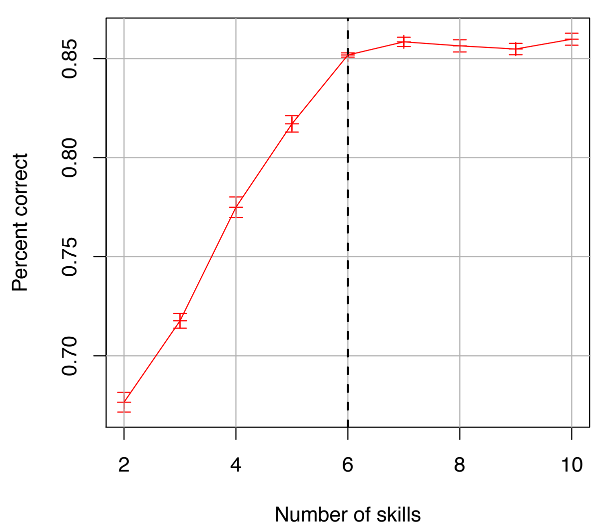


Figure 2.7 Precision of student results predictions from estimated skill matrix (equation 2.10). Error bars are the standard error of the accuracy curves. Experiment is done with simulated data with 6 skills and slip and guess values of 0.1 and 0.2 respectively.

thogonality constraint of may limit the application of the SVD method with respect to real skills and might explain some of the difference between the two methods. The skills from the synthetic data of the first experiment were independent and the Q-matrix had an homogeneous pattern for each skill, and therefore the effect of dependence between skills could not come into play.

Obviously, the study calls for more investigations. The findings from one set of data from the real world may be highly different from another set. More studies should be conducted to assess the generality of the findings. Other investigations are called for to find ways to improve these methods and to better understand their limits when faced with real data. In particular, we need to know at which level of noise from guess and slip factors do the methods break down, and what is the ratio of latent skills to data set size that is critical to avoid over-fitting of the wrapper method.

One improvement that can be brought to the wrapper method is to use a cross validation to derive the skills matrix. This would require the use of two sets of items, one for testing and one for assessing the student's skills. This comes at the cost of a greater number of items, but it avoids the problem of over-fitting that leads to accuracy increases.

2.10 The refinement of a Q-matrix

Validating of the expert defined Q-matrix has been the focus of recent developments in the field of educational data mining in recent years (**?????**). In this section we compare three data driven techniques for the validation of skills-to-tasks mappings. All methods start from a given expert defined Q-matrix, and use optimization techniques to suggest a refined version of the skills-to-task mapping.

2.10.1 Q-matrices validation techniques

Two techniques for Q-matrix validation surveyed here rely on the DINA and DINO models, whereas one relies on a matrix factorization technique called ALS. We briefly review each technique below before describing the experiments.

de la Torre (2008) The method defined by de la Torre (**?**) searches for a Q-matrix that maximizes the difference in the probabilities of a correct response to an item between examinees who possess all the skills required for a correct response to that item and examinees who do not. It first uses the DINA model and an EM algorithm to estimate the slip and guess parameters, after which it can calculate the respective probabilities. The difference between the two probabilities represents an item discrimination index: the greater the difference between the probability of a correct response given the skills required to the probability given missing skills, the greater the item is discriminant. Therefore we can consider that the method finds a Q-matrix that maximizes item discrimination over all items.

Chiu (2013) Chiu defines a method that minimizes the residual sum of square (RSS) between the real responses and the ideal responses that follow from a given Q-matrix (**?**). The algorithm adjusts the Q-matrix by choosing the item with the worst RSS over to the data, and replaces it with the one has the lowest RSS, and iterates until convergence.

ALS The (ALS) method is defined in (**?**). Contrary to the other two methods, it does not rely on the DINA model. Instead, it decomposes the results matrix $\mathbf{R}_{m\times n}$ of m items by n students as the inner product two smaller matrices:

$$\mathbf{R} = \mathbf{QS} \tag{2.11}$$

where $\mathbf{R}$ is the results matrix (m items by n students), $\mathbf{Q}$ is the m items by k skillls Q-matrix, and $\mathbf{S}$ is the mastery matrix of k skills by n students.

2.10.2 Methodology and data sets

The two first methods, de la Torre (2008) (**?**) and Chiu (2013) (**?**), have been shown to perform well on artificial data. On real data, the performance is more blurry. The ALS factorization method (**?**) has only been tested on one real data set. But the methodologies used to validate all three techniques in each respective study vary considerably and do not allow for a proper comparison.

To validate and compare the effectiveness of each technique for refining a given Q-matrix, we follow a methodology based on recovering the Q-matrix from a number perturbations: the binary value of a number of cells of the Q-matrix is inverted, and this "corrupted" matrix is given as input to each technique. If the technique recovers the original value of each altered cell, then we consider that it successfully "refined" the Q-matrix.

A total of 30 perturbations are randomly injected in each Q-matrices to validate a method's capacity to recover the original matrix. Note that when a single perturbation is injected, the maximum number of different perturbations is the size of the Q-matrix. Therefore, if is the size of the Q-matrix is smaller than 30, the total number of perturbations is limited to the number of cells in the Q-matrix.

The experiment is repeated for each of the 10 levels of perturbation, and for each of the 10 data sets described later. This set of experiments is referred to as a single run, and performance measures are averaged over 8 runs.

The measures of performance are the number of *true positives* and *false positives*:

Table 2.2 Data sets

	Name	Number of Skills	Number of Items	Number of Cases	Description
1.	Sim. DINA	3	9	400	Artificial data available from the (`sim.dina`) data set of the CDM package.
2.	Sim. DINO	3	9	400	Same parameters as No. 1 but using the DINO model (`sim.dino` data set).
3.	Sim. CDM DINA	3	12	4000	Artificial data generated through the CDM function `sim.din`.
4.	Sim. DCM	3	7	10000	Artificial data from chapter 9 of the book *Diagnostic Measurement* **?**
5.	ECPE	3	28	2922	Dataset from **?** in **?**
6.	Fraction	8	20	536	Tatsuoka's fraction algebra problems **?** (see table 1 in **?** for a description of the problems and of the skills).
7.	Fraction 1	5	15	536	15 questions subset of Fraction with Q-matrix defined in **?**.
8.	Fraction 2/1	3	11	536	11 questions subset of Fraction with Q-matrix from **?**.
9.	Fraction 2/2	5	11	536	11 questions subset of Fraction with Q-matrix from **?**.
10.	Fraction 2/3	3	11	536	3 skills version of Fraction 1.

— **Mean true positives**: a *true prositive* corresponds to an alteration that was injected in the input, and was correctly switched back to its original value by the method. The measure reported is the number of correctly recovered alterations averaged over the 8 runs.

— **Mean false positive ratio**: a *false positive* corresponds to a changed Q-matrix entry returned by the method, but that was not injected in the input. Hereto the mean over all perturbation runs is given.

For real data, this methodology rests on the assumption that the original matrix is better than the corrupted one, which is not necessarily the case with an expert generated Q-matrix. The expert may be wrong. However, we have no other means to inform us of the "real" Q-matrix and it is reasonable to assume that most of the cells in the Q-matrix are correct.

For synthetic data, this assumption is correct as the Q-matrix is at the source of the generation of the data, but, of course, the model behind the process may not be a reliable reflection of the real cognitive processes involved.

A total of 10 data sets are used for the validation. They are freely available from two R packages: CDM (`http://cran.r-project.org/web/packages/CDM/index.html`) (**?**) and Chiu (2013) (`http://cran.r-project.org/web/packages/NPCD/NPCD.pdf`). Table 2.2 contains a short description of each data set. Note that for the last six data sets, the source data is the same, but different Q-matrices are defined over them and subsets of items are used in the last four: the fraction data set data is used to create four variations through subsets of questions and alternative Q-matrices (Fraction 1, Fraction 2/1, Fraction 2/2, and Fraction 2/3). The artificial data sets are generated from the well known DINA and DINO models.

For obtaining the results of the de la Torre (2008) method, we used the R implementation found in the CDM package (**?**). A DINA model and parameter estimation is first built with the default arguments to the `din` function, and fed to the `din.validate.qmatrix` function to obtain a refined version of the Q-matrix. For the results of the Chiu (2013) method, the R NPCD packaged is used (function `Qrefine`).

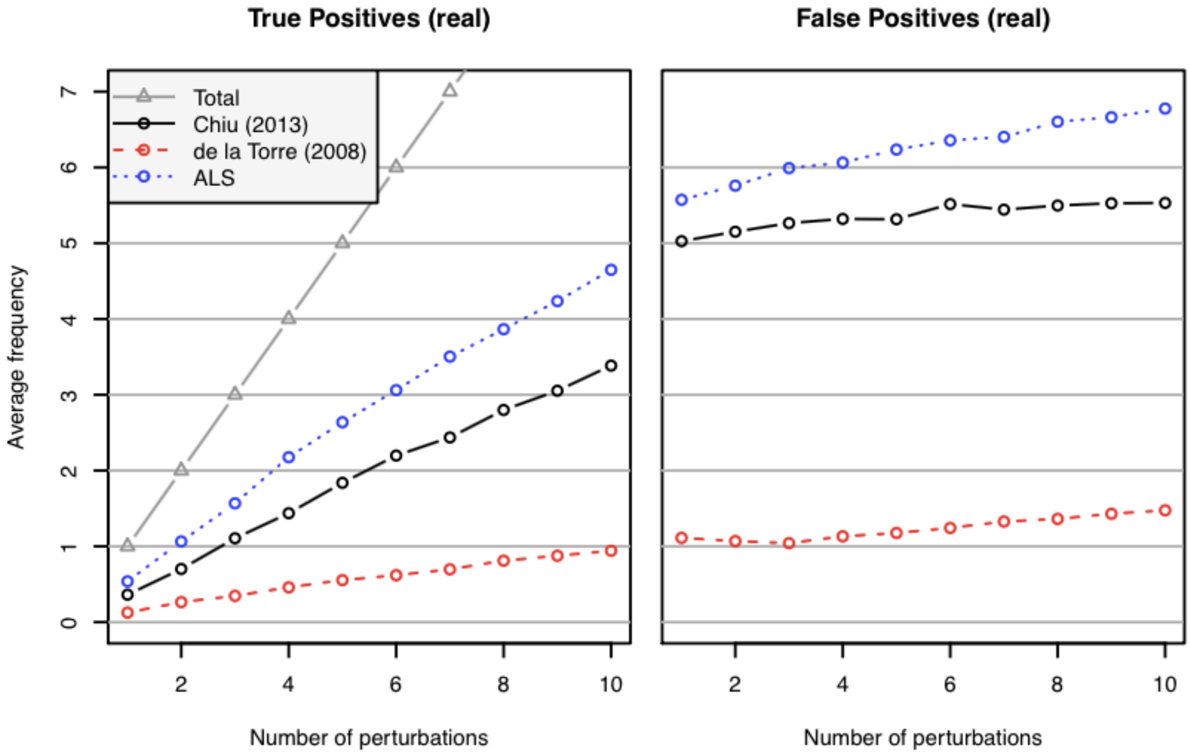
2.10.3 Results

The three methods are evaluated over the ten data sets and for ten runs. Each run is conducted over a set of 30 different random combinations of perturbations, from 1 up to 10 perturbations. For the 1-perturbation condition, the total number of possible combinations is the size of the Q-matrix itself.

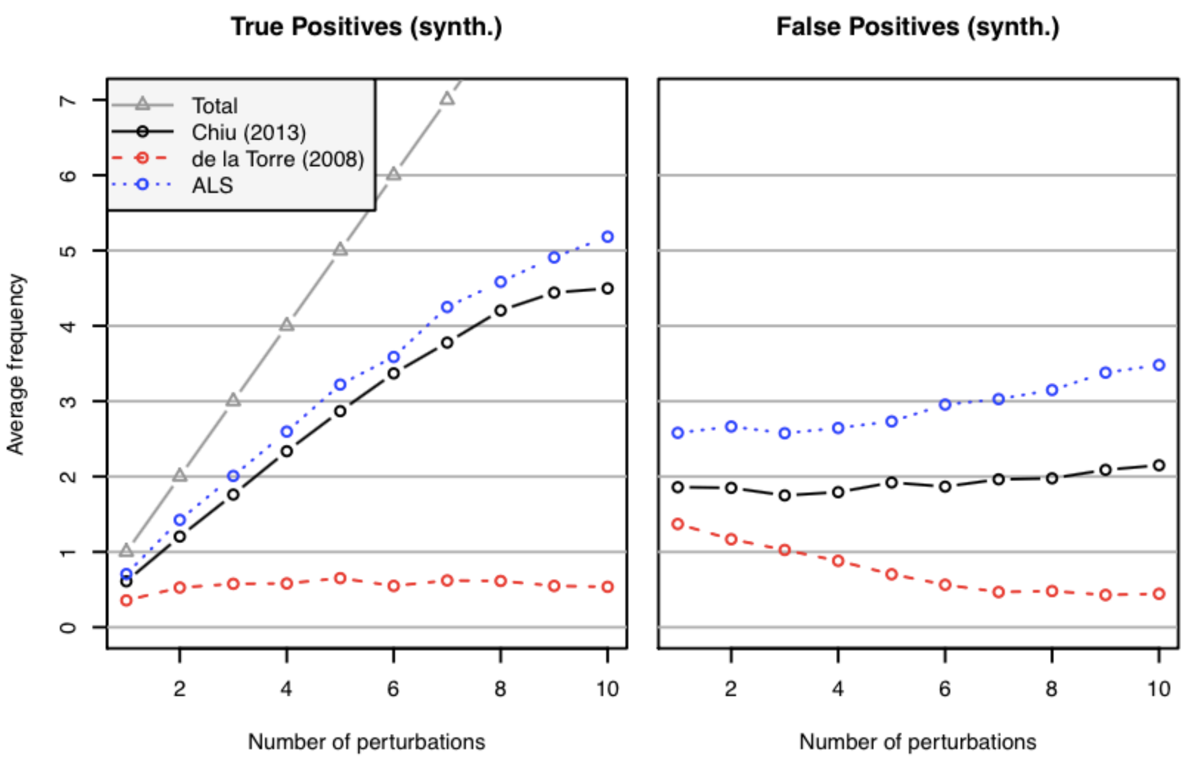
2.10.4 Recovery rates by the number perturbation

Figure 2.8(a) shows the average recovery rate of each method as a function of the number of perturbations. Recoveries are labeled "True Positives" (TP) whereas changes introduced by a method, but which do not correspond to perturbations, are labeled "False Positives" (FP). The top two graphs show the averages of the 6 real data sets, whereas the bottom graphs show the averages for the 4 synthetic data sets. Averages are computed over the 30 perturbations runs (or less if the Q-matrix has fewer cells). The "Total" line is shown to visually indicate the maximum that can be reached by a TP curve.

The ALS method shows the greatest ability to recover alterations, but at the cost of a higher rate of FP: changes that do not correspond to perturbations. It is followed closely by the Chiu (2013) method. The de la Torre (2008) method has a very low rate of recovery (TP) that make it impractical. In general, the ALS and Chiu (2013) methods recover about 2/3 of the perturbations for synthetic data, and this rate falls to 1/2 for real data with ALS, and about 1/3 for Chiu (2013). For real data, the number of FP is around 5 for Chiu (2013) and around 6 for ALS, whereas it is respectively 2 and 3 for synthetic data. The relative performance of Chiu (2013) with respect to ALS is



(a) Average recovery rate by number of perturbations (real data)



(b) Average recovery rate by number of perturbations (synthetic data)

Figure 2.8 Average recovery rate by number of perturbations (real and synthetic data).

Table 2.3 Results by individual data set at 1 and 4 perturbations

	Mean True Positives		Mean False Positives		TP/FP		
	ALS	**Chiu (2013)**	**ALS**	**Chiu (2013)**	**ALS**	**Chiu (2013)**	**Ratio**
Synthetic @ 1 perturbation							
Sim. DINA	0.70 (0.00)	0.55 (0.05)	2.15 (0.00)	0.28 (0.05)	0.33	1.95	0.3
Sim. DINO	0.48 (0.00)	0.28 (0.03)	5.59 (0.00)	5.30 (0.17)	0.09	0.05	1.8
Sim. CDM DINA	0.93 (0.01)	1.00 (0.00)	0.00 (0.00)	0.00 (0.00)	∞	∞	-
Sim. DCM	0.43 (0.00)	0.14 (0.00)	0.00 (0.00)	0.16 (0.02)	∞	0.89	∞
Real data @ 1 perturbation							
ECPE	0.65 (0.05)	0.36 (0.03)	9.81 (0.18)	17.81 (0.16)	0.07	0.02	3.5
Fraction	0.56 (0.07)	0.46 (0.06)	5.83 (0.19)	2.20 (0.27)	0.10	0.21	0.5
Fraction 1	0.65 (0.05)	0.49 (0.08)	5.25 (0.17)	3.14 (0.27)	0.12	0.16	0.8
Fraction 2/1	0.52 (0.03)	0.42 (0.03)	4.56 (0.09)	5.92 (0.12)	0.11	0.07	1.6
Fraction 2/2	0.61 (0.09)	0.41 (0.04)	4.14 (0.18)	0.59 (0.12)	0.15	0.71	0.2
Fraction 2/3	0.36 (0.02)	0.23 (0.04)	9.39 (0.07)	5.37 (0.20)	0.04	0.04	1.0
Synthetic @ 4 perturbations							
Sim. DINA	2.55 (0.23)	1.88 (0.19)	2.55 (0.34)	0.95 (0.28)	1.00	1.99	0.5
Sim. DINO	1.69 (0.15)	1.12 (0.13)	5.21 (0.19)	4.43 (0.19)	0.32	0.25	1.3
Sim. CDM DINA	3.57 (0.11)	4.00 (0.00)	0.18 (0.07)	0.00 (0.00)	19.75	Inf	0.0
DCM	2.15 (0.25)	0.53 (0.13)	1.01 (0.13)	0.57 (0.12)	2.13	0.94	2.3
Real data @ 4 perturbations							
ECPE	2.47 (0.23)	1.49 (0.21)	10.59 (0.32)	17.35 (0.20)	0.23	0.09	2.5
Fraction	2.28 (0.14)	1.86 (0.15)	7.93 (0.29)	2.74 (0.23)	0.29	0.68	0.4
Fraction 1	2.56 (0.08)	1.95 (0.25)	5.48 (0.25)	3.98 (0.23)	0.47	0.49	1.9
Fraction 2/1	2.19 (0.22)	1.80 (0.20)	4.92 (0.27)	5.30 (0.28)	0.44	0.34	1.3
Fraction 2/2	1.90 (0.17)	1.04 (0.16)	4.32 (0.35)	1.91 (0.28)	0.44	0.54	0.8
Fraction 2/3	1.69 (0.15)	1.41 (0.13)	8.20 (0.23)	5.40 (0.24)	0.21	0.26	0.8

better for synthetic data and this might be explained by the fact that the data generation process is directly based on the DINA model.

A common pattern across methods is the relatively stable number of FP as a function of the number of perturbations. ALS does show an increase of close to 1 FP between 1 to 10 perturbations, whereas the increase for the Chiu (2013) and de la Torre (2008) methods is closer to 1/2 for real data, and even less for synthetic data (in fact it is -1 for de la Torre (2008)). As a result, the rate of TP over FP increases with the number of perturbations.

2.10.5 Recovery rates by data set

The means of TP and FP provide the general trends, but the question remains whether these trends are systematic across each data set. To investigate this question we look at the performance details at 1 and at 4 perturbations. They are reported in table 2.3 for the ALS and Chiu (2013) methods. The de la Torre (2008) method is ommitted for brevity and because its performance is much worst than the other two.

The TP and FP means over the 30 perturbation runs of each individual data set is reported along with the standard deviation within parenthesis. We also report the ratio TP/FP. The “Ratio” column corresponds to the ALS TP/FP ratio over the Chiu (2013) TP/FP ratio: a value greater than 1 indicates that the TP/FP ratio is in favor of ALS over Chiu (2013), and the opposite occurs if the value is less than 1. Based on this general indicator, we can conclude that the ALS and Chiu (2013) have a similar performance, but the advantage varies considerably across data sets and can go to either of each method.

Also notable is that the two methods show close to perfect performance at a single perturbation for large simulated data sets (Sim. CDM DINA and DCM). However, at the opposite, they have a weak TP/FP ratio for the ECPE and Fraction 2/3 data sets (from 0.02 to 0.07). This implies that only between approximately 1 in 50 to 1 in 15 of the proposed Q-matrix changes actually corresponds to perturbations. This may prove insufficient for practical purposes, but then again we have to remind ourselves that not all FP may be considered invalid proposition for changes. Furthermore on the positive side, the TP/FP ratios at 4 perturbations are more in the range of 1/5 for these same data sets which is a more encouraging.

2.10.6 Discussion

The experiments conducted in section 2.10 confirm that all methods can recover an altered Q-matrix, as shown in previous work **???**. But the comparison of their performance over a number of data sets, and based on a common measure of performance, reveals wide differences.

The ALS matrix factorization technique shows a greater ability to recover alterations than the other two techniques. For real data, the differences are highly in favor of this method. Even for data sets made of artificial data generated from the DINA model, with which the two other techniques rely upon for refining the Q-matrix, the ALS factorization performance is comparable or better than the other two techniques.

In addition to revealing the differences of performance among the approaches, the methodology for comparing these techniques is one contribution of this research. Previous work rely on artificial data and parameter fit measures, i.e. the difference between an estimated and an the original value of the parameter used to generate the artificial data, as a performance indicator. However, this approach remains feasible only for models which share the same parameters. The principle of measuring the proportion of single alterations that are correctly restored is applicable to all models, regardless of their specific parameters.

This methodology could be enhanced to include sets of more than one alteration, but the number of experimental runs increases exponentially, $(m \times n)^a$ (where $m \times n$ is the size of the Q-matrix and a is the number of alterations). For larger number of alterations, sampling would be needed to contain the growth of runs.

In addition to extending the size of the number of alterations, future work should also extend the comparison to more techniques such as **??**. Finally, we re-iterate the need for open access to the data and the code used in such studies. This particular study was highly facilitated by the CDM **?** and NPCD packages which provided both the code and the data.

2.11 Improving matrix factorization techniques of student test data with partial order constraints

The very first contribution of this thesis was improving the Matrix factorization techniques of student test data with partial order constraints. In particular, we want to address this question: can a Partial Order Knowledge Structure (POKS) be used to guide matrix factorization algorithms and lead to faster or better solutions to latent skills modelling?

One avenue to improve over current matrix factorization models is to adapt existing algorithms to the specific nature of the domain data. In particular, student performance data is known to be constrained by prerequisite relations among skills or knowledge items **??**. This constraint can substantially reduce the space of factorization, both for the purpose of assessing student skills and for mapping items to skills. The objective of this research was to explore how one type of constraints, known as Partial Order Knowledge Structures (POKS), can lead to better factorization techniques for the purposes mentioned.

The very first solution for this improvement is to develop a new factorization algorithm based on the POKS constraints. The idea is to change the cost function of the standard NMF function. As described in the previous section, standard NMF algorithm works based on the cost function in equation 2.2. Adding an other value to this cost function can lead to a new equation 2.12.

$$\|\mathbf{R}-\mathbf{QS}\|^2+\kappa(\mathbf{O}\hat{\mathbf{R}}) \tag{2.12}$$

where κ is a normalizing constant and $\mathbf{O}\hat{\mathbf{R}}$ is the product of the Ordinal incidence matrix gained from POKS algorithm and the expected results matrix obtained from product $\hat{\mathbf{Q}}\hat{\mathbf{S}}$.

The second term of this formula is a penalizing factor based on the POKS constraints. For simplicity we wont explore details of implementation for this part of cost function.

In order to test the hypothesis we run a simple experiment to see if POKS can give new information which could improve NMF results or not. The methodology is to test NMF, POKS and the combined method with a Bayesian and a Linear generated dataset. The reason for this experiment is to verify if the model behind a dataset can affect the performance of different techniques. Table 2.4 shows the result of this experiment. Note that the basic concern of this study is working with noisy data. It was shown that NMF perfectly recovers the Q-matrix the with non noisy data, but as some noise like slip or guess was added to the data, the result differs from what was expected (**??**). In this study we want to use another source of knowledge to make an improvement. We should make sure that POKS would be correctly derived from both noisy and non-noisy data. In order to generate a noisy data we changed 50% of values for on particular item (say first item) which can represent noise such as slip and guess. The results of experiment showed that NMF could prefectly recover the expected Q-matrix from a linear generated dataset while POKS not even recovered the relations on the noisy item but it presented some false relations between items. On the other hand the estimated Q-matrix from the combined method was not as perfect as the Q-matrix derived from NMF.

The conclusion that was made out of this hypothesis is: There is a correlation between the underlying model of a dataset and predictive performance of a skills assessment technique. POKS as a Bayesian model can predict the Partial Order parameters of a Bayesian generated dataset better

Table 2.4 Results of running NMF, POKS and combined methods on Linear and Bayesian generated datasets

Parameters	Bayesian Generated	Linear Generated
Q-matrix by NMF	Noisy	Perfect
Knowledge Structure by POKS	Perfect	Noisy
Q-matrix by Combined method	Noisy	Noisy

than a linear generated dataset. The reverse is true for factorization techniques on linear generated dataset.

This experiment showed that combining two different models can not necessarily improve one of them which means that the performance is really depends on the underlying model of the dataset. This is where we created the second contribution to assess the model fit with comparing the predictive performance of synthetic vs. real dataset.

CHAPTER 3
MODEL FITTING AND SIMULATED DATA

3.1 Model Fitting

Model fitting is the task of selecting a statistical model for a given data from a set of candidate models. Both data and the set of candidate models can involve the design of experiments that the data collected is well-suited to the problem of model selection or the candidate models best describe the data among a large number of models. Generally Model fitting consists three steps :

— Prediction : that takes a set of parameters and returns a predicted dataset

— Error function : that gives a measure to represent the difference between the data and the model's prediction for a given set of parameters

— Minimizing the error function : finding a set of parameters that minimize the error function

The best fit is the model that is most likely to have generated the data.

3.1.1 Approaches

The simplest way to address the first step is providing the predictive performance accuracy which is a basis for comparing models. Models with higher predictive accuracy yield more useful predictions and are more likely to provide an accurate description of the ground truth. Cross validation is also a straight forward and easy to understand approach for estimating predictive performance. In this research we used 10 fold cross validation to get the predictive accuracy.

3.1.2 Measures

There are different measures to represent the degree of similarity in step two and this is usually either the sums of squared error (SSE) or maximum likelihood. Model fitting is also called as an optimization method where each model in the set of candidate model optimized the error function (usually likelihood function) to find the best fitting model.

Other measures such as Euclidean distance, cosine and correlation can also be used as a error function to compare the data and the model's prediction. The best fitting parameters can depend strongly on your choice of error function. In our study we use both 0/1 loss function as an error function and correlation as a measure of similarity.

The maximum likelihood function selects a set of values for the model parameters that maximizes the likelihood function which also maximizes the agreement of the selected model with the observed data. Likelihood function is also called inverse probability which is a function of the parameters of a statistical model. This allows us to fit many different types of models.

In some context the variables can be hold constant while fitting. In this case the fitting function should preform with a subset of parameters. For example in the context of educational data mining the Q-matrix is known as a parameter which is defined by an expert. Suppose that you know the Q-matrix and we just want to let the other parameters (such as skills mastery matrix) vary. We do this by only listing "skills mastery matrix" as a free parameter to fit.

The notion of *goodness of fit* represents how well a model can account for observed data. For example, student test results can be accounted for by the ability of each student, and by the difficulty of each question, its discriminative factor, and by a guessing factor. This is the basis of the 3-parameter logistic model IRT-3PL **?**. Given some training data to estimate the model parameters, and a partial observation of a test data set, held out responses can be "predicted". The difference between the predicted and the actual responses represents the residual error, which is considered a measure of the goodness of fit. And the lower the goodness of fit, the closer is the model considered to the ground thruth.

A number of factors can affect the amount of residual error. The capacity of an algorithm to estimate the model parameters for a given data set is often critical. Local optima, biases, and large variance can result in estimates that are far from the best ones **?**. Models themselves can yield widely different performances under different circumstances. Some are more robust under small data sets. Typically, complex models will require large data sets, but can sometimes lead to much better performance than simpler ones if they are fit for the data.

For these reasons, a model may be "fit to the data", and yet it may underperform compared to others when residual error is used as the sole indicator of the goodness of fit. The residual error is always measured for a given data sample, and to obtain a reliable estimate of the goodness of fit, data samples that cover the space of factors that can affect parameter estimates and model performances would need to be gathered. Oftentimes this is impractical.

3.1.3 Dynamic vs. static data

The knowledge tracing model (**?**) is one of the best known models that has been widely used to model student knowledge and learning over time. It has four parameters where two of them (prior and learn) are knowledge parameters for each skill. This is a standard model of knowledge tracing that consider the fact that students are learning over time. Tensor rank decomposition is also a general model that is used a time-based datasets. In multilinear algebra, the tensor rank decomposition may be regarded as a generalization of the matrix singular value decomposition

(SVD) to tensors. SVD is a well known model in EDM where has been used in skill assessment approaches (**?**).

The difference between tensor based methods and static methods is in the complexity of them. Tensor based approaches like BKT and canonical polyadic decomposition (CPD) are modeling a dynamic learning environment where static methods such as POKS, NMF, DINA/DINO and IRT are modeling a snapshot of student test performance. All data sets in this research are considered *static* in the sense that they represent a snapshot of student test performance data. This corresponds to the assumption that the student has not mastered new skills during the process of assessment, as we would expect from data from learning environments. This assumption is common to all models considered for this study.

The following sections are describing two recent works that have been done for model fitting in EDM:

3.2 On the faithfulness of simulated student performance data

Michel Desmarais in (**?**) introduced an approach to investigate the faithfulness of different methods of generating simulated data by comparing the predictive performance of POKS over real vs. simulated data. The parameters for simulated datasets are set to represent those of the real data. The faithfulness of the synthetic data is dependent to its performance. The more similar is the performance of real vs. simulated data is, the more faithful it is to represent the real data.

In general there are three approaches to validate the accuracy of a cognitive diagnostic model without a direct measures of skills mastery:

- Indirect and independent measures of skills mastery: in these methods some expert defined skills mastery mappings are created based on the students answers to a test. Vomlel (**?**) and De La Torre (**?**)asked experts to define these matrices for two datasets. One of the weakness of this approach is that different experts may introduce different skills or different mappings.
- Predict based on observed items only: This method does not try to predict skills mastery mappings but it predicts based on a observed set of items. In our research we are using this method as a part of our methodology.
- Generating simulated data : This is the method that Desmarais (**?**) used in his work. They used a set of predefined parameters to generated a result matrix based on a specific model.

3.2.1 Simulated data models

In (?) they used POKS as the student model which is a Bayesian approach to cognitive modeling. They take the closest performance of this model over a real vs. synthetic data. For generating synthetic datasets they used four models:

- Expected outcome based on Marginal Probabilities: This is the expected value for the probability of item outcome which is a function of marginal probabilities of item success rate and student scores.
- Q-matrix Sampling: In their experiment, conjunctive model of Q-matrix is used where skills of an item must be involved in order to correctly answer that item.
- Covariance Matrix:Synthetic test result is generated based on a technique that preserve covariance (correlation) among items. This method is usually used in Monte Carlo simulations. In this particular study this method reflects correlation among student response patterns that derived from items with similar difficulties and same skills set.
- Item Response Theory: they used 2 parameters logistic regression IRT model to generate simulated data.

3.2.2 methodology

Once the simulated data is generated base on the four models they (?) train POKS student model over both real and synthetic datasets. For validation of the accuracy of the simulated dataset they compare the predictive performance across each condition.

3.2.3 Real Datasets

It is important to choose a good dataset for this simulation, they used two datasets which are in math and one of them has small number of samples and the other one has big number of items. The details of these datasets are in bellow :

Dataset	Number of		Average	Item
	Items	Students	Success rate	Success rate Variance
Unix	34	48	53%	1/34 to 34/34
College Math	59	250	57%	9/59 to 55/59

3.2.4 discussion

First let us summarize their conclusions and then propose our discussion. The following items are their conclusion :

— This experiment need to be expanded since it was based on a single student model which is POKS and also other models of simulated dataset should also be used in this evaluation.

— The expected marginal probability do not appropriately reflect the underlying ground truth of the real datasets

— For the first dataset (*Unix*) IRT was the best representative for the underlying structure of the real data where the predictive performance of real data was 77% and IRT generated data was 80%

— For College math data, the synthetic data generated based on the *covariance* method shows a performance which is closer to the performance of real data than others. The accuracy gain is 40% for real data when it is 37% for covariance generated data.

— Validating the faithfulness of student models requires assessing parameters of those models to replicate real data characteristics.

— simulated data from the 2 parameter IRT model can appropriately reflect some dataset characteristics but not with equal faithfulness for all datasets.

In this thesis we use 7 student models for generating datasets and also measuring the predictive performances that include range of models from zero skills to multi-skills. Somehow our work can be an extension to (**?**). Obviously one of the reasons that IRT generated data shows a good similarity with real data performance is that the performances are over POKS model which shows closest performance to POKS model(The details will be discussed later in 5) where the other models are linear and multi-skills.

To summarize the difference between our experiment and (**?**) we can say that our work is a kind of extension to this research. Both of them are comparing the behavior of different datasets with different underlying structure. The difference is in the number of predictive performance models. (**?**) uses only one technique to fit a model but in our work we compare a set of models which create a signature and those datasets that have similar signature can reflect similar characteristics.

3.3 simulated data to reveal the proximity of a model to reality

The next recent work (**?**) is about distinguishing between a synthetic data and a real data. This work is an extension to their previous work (**?**) where they used BKT model to generate synthetic dataset for two real dataset that correspond to the characteristics of the real data. They found similarities between the characteristics of the simulated and real datasets. The results indicate that it is hard to set real and synthetic datasets apart. The idea of this research (**?**) is about the goodness of a model for a real dataset which indicate that if it is easy to set real and synthetic data apart then the model is not a good representative of the real data otherwise the model is indeed authentic representation of the reality.

3.3.1 Methodology

They used Bayesian knowledge tracing model to calculate log likelihood with a grid search of four parameters : initial(prior knowledge), learning rate, Guess and slip. The two first parameters are knowledge parameters and the second two parameters are performance parameters. The simplest form of BKT which is used in this experiment considers a single set of prior knowledge and learning rate for all students and an equal slip and guess rates for all students. To make a comprehensive comparison they used 42 datasets which are groups of Learning Opportunities(GLOPs) generated from the ASSISTments platform. Problem set and number of examinees vary for each dataset which consist of 4 to 13 questions answered by 105 to 777 students. In addition they created two synthetic datasets for each dataset that the parameters for synthetic datasets are set to represent those of the real data with exact same number of samples and items.

The methodology consist of four parts :

- Calculating a best fitting parameters for all 42 real datasets
- Creating two different simulated dataset with the founded parameters and the same number of students and items
- Calculating the log likelihood of the parameters space for both real and syn. datasets
- Comparing the log likelihood gradient of Synthetic vs. Real data

The comparison in the last step of the methodology is made by visualizing a 2D log likelihood heatmap with two parameters plot where the other two parameters were fixed to the best fitting values. The similarity of the heatmap of the LL matrices of the real data and the two simulated data is a measure for model fitting in their experiment. The more they look similar the more the model fits the real data. They proposed two methods to address the degree of similarity:

- Euclidean distance : The Euclidean distance between the real dataset parameters and synthetic dataset parameters was compared to the distance of two synthetic dataset parameters. In conclusion if the distance of two synthetic parameters are smaller than the distance of real and synthetic parameters then the model is a goof fit for the data otherwise it can be improved.
- Log likelihood distance : The max log likelihood distance between the two synthetic datasets was compared to the max log likelihood distance between the real and synthetic datasets.

CHAPTER 4
SYNTHETIC DATA GENERATION

Simulated data has been increasingly playing an important role in EDM (?). Even in natural science domain, it is usually used as a way of evaluating its underlying model. As an example (?) used simulated data for disease outbreak detection where simulated data is generated from a hypothesized model of a phenomena and if there exists similarities between simulated data and real data observed in the nature, it serves as an evidence for the accuracy of the model and it may be possible to claim that the model provides an authentic explanation of the system.

Generally to validate a proposed approach, simulated data is the best candidate because all the parameters could be predefined. Once the simulated data is generated with predefined parameters, the model can be trained over the generated data. This may be the strongest benefit of synthetic data in model fitting.

The framework of this study relies on synthetic data. We rely on simulated data for all experiments at the first place. Although it lacks the external validity of real data, it remains the most reliable means of obtaining test results data for which the underlying, parameters such as latent skills structure is perfectly known. Every model studied here can be considered *generative*, to the extent that they can generate synthetic data. The data generation process of each model share common parameters which are: number of items, number of students, total success rate and data distribution but other parameters vary according to different techniques. The generation process for each model is described below.

4.1 POKS

The method that generates synthetic dataset based on the POKS model requires a knowledge structure (KS). In this process the KS can be obtained from a real data set which allows us to make a better comparison of the results. It can also be generated as a random KS. The graphical demonstration of knowledge structure is a directed tree which shouldn't have any loop, therefore in the random generation of KS this issue should be considered otherwise it violates the definition of KS. There are other input parameters which can make the random generation more specific like the number of links and number of independent trees in a KS. Applying these parameters will change the item variance in the test result matrix. In order to avoid loops in knowledge structure graph, we create an upper triangular adjacency matrix with random links. Once the KS adjacency matrix is created , we can assign values to each item that contributes with the initial odds of each node and odds ratio between a pair of nodes.

The approach for assigning values to each sample is using probabilities for each node and a ratio for each pair for nodes which represent the strength of the link. Based on the knowledge structure that has been picked we can also assess a set of initial probability for each node where a parent node gets a lower initial probability than its children. Inference in the POKS framework to calculate the node's probability relies on standard Bayesian posteriors under the local independence assumption. The probability update for node H given E_1,... E_n can be written in following posterior odds form :

$$O(H|E_1, E_2, ..., E_n) = O(H) \prod_i^n \frac{P(E_i|H)}{P(E_i|\overline{H})} \tag{4.1}$$

where odds definition is $O(H|E) = \frac{P(H|E)}{1-P(H|E)}$. If evidence E_i is negative for observation i, then the ratio $\frac{P(\overline{E_i}|H)}{P(\overline{E_i}|\overline{H})}$ is used.

All the parameters containing Partial Order structure, initial odds and odds ratio can also be obtained form a real dataset. For the case that these values are not predefined we need to assign random values with respect to the defined partial order structure. Once these values are defined, we can pick a node to sample with it's initial odds and consequently update it's neighbors' odds with equation 4.1. This process could be continued until there exists no node to sample.

The only way to control the total test result successrate and student/item score variance is to apply changes on the initial odds of nodes where we use for sampling student test result. For student scores variance, for each student the initial odds can be scaled such that the distribution of the initial odds stays the same for example we can double all the initial odds to represent a student with higher success rate. Changing the initial odds that follows a specific distribution will create a dataset in which the item variance is following that distribution. For overall successrate the initial odds can be scaled independent from students or item perspective, for example tripling all initial odds for all students will result in a higher success rate than the default values.

4.2 IRT

Equation 4.2 shows the probability of a student given the ability of θ to success item j which has the difficulty of b_j and discrimination of a_j based on IRT-2PL model. To generate a dataset that follows this model, we need to generate a sample of students with different abilities and items with different difficulties and discriminations.

$$P(X_j = 1 \mid \theta) = \frac{1}{1 + e^{-a_j(\theta - b_j)}} \tag{4.2}$$

$$-4 < \theta < 4$$

$$-3 < b_j < 3$$

Students ability to answer questions and item difficulty are generated by a normal distribution with the mean of 0 and standard deviation of 1.25 with respect to their boundaries. Discrimination (slope or correlation) representing how steeply the rate of success of individuals varies with their ability. In IRT-2PL, the values for discrimination are following a Poisson distribution with lambda parameter set to 10 that kept most values between 0.5 and 3 . Note however that for the purpose of synthetic data generation, we rely on the more general IRT-2PL model in order to make this data more realistic.

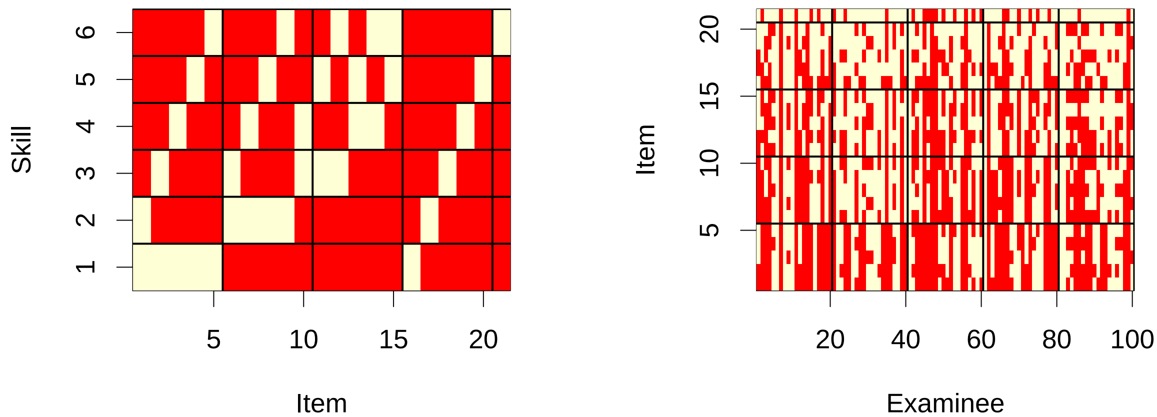
Item discrimination is a parameter that is learnt from training set. A perfect dataset that doesn't have any noise estimates this parameter with extremely large or small values(for example 300 or -300) which results in an unrealistic outcome prediction. For this purpose we add small amount of noise to prevent this condition.

4.3 Linear NMF Conjunctive

The very first step to generate simulated test result for linear models is to define a Q-matrix that maps k skills to n items. The Q-matrix can be an expert predefined matrix or a random generated matrix. In the case of unavailable predefined Q-matrix, we defined a Q-matrix that provides all the possible combination of k skills with a maximum of Max skills per item, and at least one skill per item. A total of $\sum_{k=1}^{Max} \binom{n}{k}$ items span this space of combinations for example 21 items for 6 skills and maximum 2 skills per item. This matrix is shown in Figure 4.1. Items 1 to 15 are two-skills and items 16 to 21 are single-skill. Once the Q-matrix is created we can randomly replicate or eliminate some items to adjust the number of items to the desired number.

There are two ways to apply item variance in the simulated data and both of them are based on manipulating the values of the Q-matrix. Applying skills difficulty on skills would transfer the difficulty to items that have this skill. The other method is to consider the same weight for skills difficulty but controlling the item variance by assigning different number of skills to items. For example items with 1 skill would become easier to answer comparing to items with two or more skills where skills difficulty is the same for all skills.

The second step is to create a student skills mastery matrix which maps k skills to m students. In terms of ability for examinees we assigned random values to skills for students but student variance show up as the variance in number of skills across examinees. At the same time we can apply the overall success rate on the skill mastery matrix using a threshold to discrete the assigned values in



(a) Q-matrix of 6 skills

(b) Synthetic data of 100 students with 10% slip and 20% guess factor

Figure 4.1 Q-matrix and an example of simulated data with this matrix. pale cells represent 1's and red ones represent 0's.

skills mastery.

Once the Q-matrix and Skills mastery matrix is created we can produce the test result matrix with equation (2.5). The last step is to add slip and guess factors which are set as constant values across items.

A sample of the results matrix is given in figure 4.1 where pale cells represent a value of 1 and red cells are 0. Examinee ability shows up as vertical patterns, whereas skills difficulty creates horizontal patterns. As expected, the mean success rate of the 2-skills items 1 to 15 is lower than the single skill items 16 to 21.

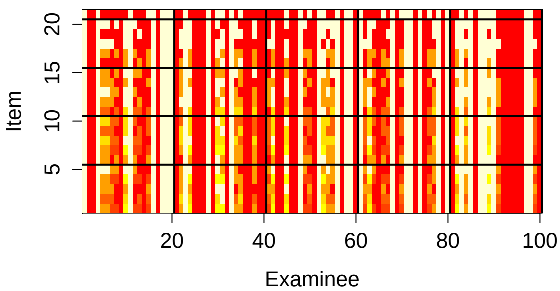
4.4 Linear NMF Additive

The process to create synthetic data based on additive type of Q-matrix is almost the same as Conjunctive one. The difference is on the interpretation of the Q-matrix that changes the step where the result matrix is producing.

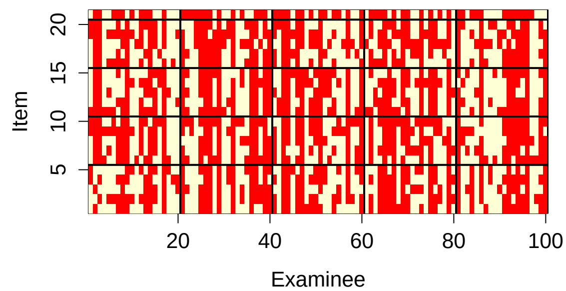
For this case each cell in the Q-matrix should be normalized on the bases of items. Although each skill has a specific weight to success an item but in our experiment we consider equal weight for all skills of an item. For this purpose all the values assigning to each item in Q-matrix should be divided by the number of involved skills for that item.

$$
\begin{array}{cc}
 & \text{items} \\
\text{skills} & \begin{bmatrix}
0.5 & 0.00 & 0.25 & 0.33 & 0.33 & 0.33 & 0.00 & 0.5 & 0.5 & 0.5 & 0.0 & 0.0 & 0.0 & 1 & 0 & 0 & 0 \\
0.0 & 0.33 & 0.25 & 0.33 & 0.33 & 0.00 & 0.33 & 0.5 & 0.0 & 0.0 & 0.5 & 0.5 & 0.0 & 0 & 1 & 0 & 0 \\
0.0 & 0.33 & 0.25 & 0.33 & 0.00 & 0.33 & 0.33 & 0.0 & 0.5 & 0.0 & 0.5 & 0.0 & 0.5 & 0 & 0 & 1 & 0 \\
0.5 & 0.33 & 0.25 & 0.00 & 0.33 & 0.33 & 0.33 & 0.0 & 0.0 & 0.5 & 0.0 & 0.5 & 0.5 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}
$$

(a) Additive Q-matrix



(b) Raw Result matrix



(c) Discretized result matrix with 20% slip and 10% guess factor

Figure 4.2 Additive model of Q-matrix and Corresponding synthetic data

Figure 4.2(a) shows an additive type of Q-matrix and figure 4.2(c) is the result of cross product of this matrix to a students skills mastery matrix. Since the model is additive, there are some pale cells and the paler a cell is, the more chance a student has to success the question. In conjunctive model the result matrix is either 0 or 1.

4.5 DINA/DINO

These models are also categorized as linear models that use a Q-matrix. The Q-matrix can be predefined for a better comparison or can be synthetic. For synthetic Q-matrix we use the same method as described before. At the same time we can control the number of skills and items in generation of a Q-matrix.

Equation 4.3 requires three parameters to predict a test outcome. In our experiment we create a sequence of values for guess and slip in the range of 0 and 0.2%. Examinee's skills can be generated by a normal distribution which should match the number of skills presented in the Q-matrix. The difference between creation of skills mastery matrix in DINA/DINO and NMF is the way that skills are appearing for each student. In DINA/DINO there is a predefined set of possible combination of skills that can be used in skills mastery for each student. For example for 3 skills, this set can have maximum 8 combination. There is a distribution that is assigned to this set which defines the probability of appearance for each combination in the skills mastery matrix.

$$P(X_{ij}=1 \mid \xi_{ij}) = (1-s_j)^{\xi_{ij}} g_j^{1-\xi_{ij}} \tag{4.3}$$

We can apply total success rate during the creation of student's skills mastery matrix where students skills define the success rate of a dataset. Since these models are behaving based on a single value that represents student ability, we need to calculate an array of abilities for each item given a set of skills for an student. In DINO we use a disjunction between skills of an item and skills of a student to determine whether the student has the ability or not where for DINA a conjunction applies.

CHAPTER 5
Experimental Results

In this chapter we present a method of determining the model that is closest to the ground thruth by defining a space to characterize models based on their predictive performance. A real data set is a point in that space, and synthetic data sets associated with each model are also individual points. The model's synthetic data set that is the closest to the real data set is deemed the ground truth. Details of this framework is described later. We first review the basic concepts and the models used in this study.

5.1 Model fit in a vector space framework

The approach we propose to finding the model that best fits a data set is based on the assumption that the predictive performance of a given model will vary as a function of the data set's groud truth model, and that the relative performance between different data sets will be stable.

Let us explain this idea with the performance data in table 5.1, where the predictive accuracy of 6 well known skills models is reported against 6 synthetic data sets generated with the same models. A seventh "model" named *expected* and seventh data set named *random* are added for comparison purpose. The details of the different models and of the methodology to assess model performance are described later. For now, let only focus on this table's data.

Table 5.1 Vector space of accuracy performances

Model	**Synthetic data set**						
	Random	POKS	IRT	DINA	DINO	NMF.Conj	NMF.Comp
Expected	**0.75**	0.91	0.90	0.72	0.72	0.78	0.93
POKS	0.75	**0.94**	0.94	0.81	0.81	0.90	0.94
IRT	0.75	0.91	**0.95**	0.73	0.73	0.79	0.89
DINA	0.75	0.77	0.81	**1.00**	0.65	0.98	0.89
DINO	0.75	0.63	0.56	0.66	**1.00**	0.68	0.91
NMF.Conj	0.75	0.59	0.53	0.95	0.65	**0.95**	0.58
NMF.Comp	0.75	0.76	0.79	0.59	0.93	0.70	**0.98**

As we can expect, the diagonal (in bold face) always displays the best performance since it corresponds to the alignment of the model and the ground truth behind the data. This confirms the intuition behind the usual strategy of assuming the best performer is the model behind the ground truth. However, this is not always the case as we will see later.

The principle of the proposed approach is to use the whole column of performance as a vector to determine the closest model to the ground truth. In that respect, if columns are considered as vectors in the space of dimensions created by model performances, we can use a similarity measure to determine the closest ground truth (or a distance measure if we were to consider the columns as a point in space).

The advantage of this approach is that it does not rely on a single performance value to determine the goodness of fit, but instead on a set of performances over different models. The hypothesis is that this set of performances provides a more reliable measure of the goodness of fit of a set of models. In turns, we assume that this measure is more likely to indicate which model will perform better in general, as opposed to which models performs the best in the case of the single data set at hand.

The approach can be considered as a means to avoid a kind of local minimum, considering the best performer as a good indicator of the ground truth, but not a perfect one. Indeed, table5.1 suggests that aligning the model with the ground truth does yield the best performance, but we will show examples later that there are exeptions and that the proposed approach is better able to avoid these exceptions that would lead to a wrong conclusion if we were to rely on the best performer approach.

5.2 Skills Models

The approach we introduce to assess model fit relies on the generation of synthetic data from different skills models. We review the basics of each of these models. We focus here on the assessment of static skills, where we assume the test data represents a snapshot in time, as opposed to models that allow the representation of skills that change in time, which is more typical of data from learning environments (see (**?**), for a review of both approaches). However, the approach would generalize to dynamic data as well.

For modeling static test data, Item Response Theory (IRT) is likely the most established model. It dates back to the 1960's and is still one of the prevailing approaches (**?**). But, akin to the trend in data mining in general, many other models have been introduced in recent years. Among them is the family of models that rely on slip and guess factors (**??**), such as the DINA (Deterministic Input Noisy And-Gate), DINO (Deterministic Input Noisy Or-Gate), and other variants. Other approaches are based on the Knowledge Space theory of Doignon and Falmagne (**?**), which does not directly attempt to model underlying skills but instead rely on observable items only ((**?**); see also extentions that do include skills: (**?**)). Finally, recent models based on matrix factorization have also emerged in the last decade (**????**). They factorize the student per item results matrix into the linear product of the so called Q-matrix (skills required per question item) and the skills mastery matrix.

There exists a large array of models to represent and assess student skills. We focus here on the assessment of static skills, where we assume the test data represents a snapshot in time, as opposed to models that allow the representation of skills that change in time, which is more typical of data from learning environments (see (**?**), for a review of both approaches).

The skills assessment model we compare can be classified at a first level according to whether they model skills directly, and whether they are single or multiple skills. Then, multi-skills model can be further broken down based on whether they have guess and slip parameters, and whether the skills are considered disjunctive or conjunctive. Figure 5.1 shows this hierarchy of models.

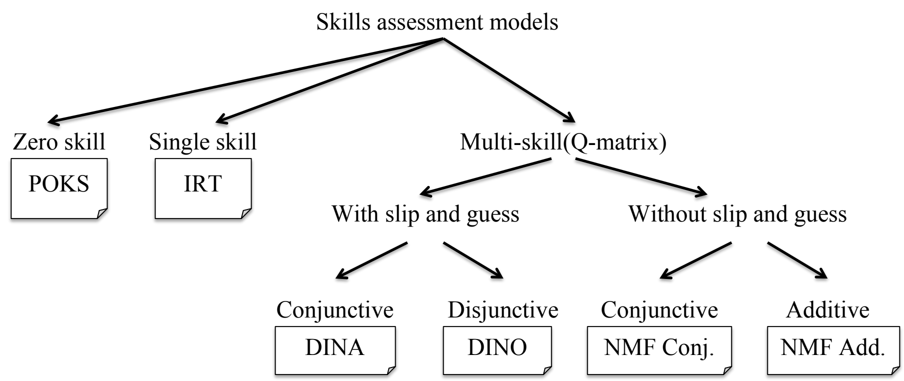


Figure 5.1 Skills assessment methods

The POKS model avoids the specification of skills alltogether by modeling the order in which items or tasks are learned. Skills can be be modeled in a second phase once these items are mapped to given skills. IRT is a well established family of models that rely on a single skill to model student performance data. Finally the multi-skill family of models rely on a mapping of item to skills, called a Q-matrix. The DINA model incorporates the guess and slip parameters and considers the all skills specified in a Q-matrix as required in order to succeed an item, whereas the DINO model requires any skills to be sufficient. The NMF conjunctive and NMF additive models are the counterpart of DINA and DINO, but they do not incorporate the guess and slip parameters, and the NMF additive model is closer to a continuum between conjunction and disjunction, as every skill mastered is considered to increase the chances of success as opposed to strict a disjunction for which any single skill is sufficient.

5.3 Methodology

In a first experiment, we focus on showing the performance of all models over synthetic and real data sets. It provides an overview of the relative performance of each model across the different synthetic data sets and across real data as well.

In a second experiment, we move focus to the central problem of this paper: classifying data sets in the performance vector space. To validate the approach, we need to rely solely on synthetic data for which we know the underlying ground truth model. A matrix such as the one in figure 5.1 is created with data sets generated from the different models, and each model performance is measured through a cross validation process. This matrix allows us to classify a data set of unknown ground truth according to a nearest neighbour approach.

5.4 Experiment 1: Performance comparison

The performance of each model is assessed on the basis of 10-folds cross-validation. The training set is used to estimate model parameters that are later used in for the test set. For each test set, a model is fed with a set of item outcomes of a student, called the observed set, and the remaining items are the predicted, or inferred ones. The breakdown of the data for cross-validation is illustrated in figure 5.2. We fixed the number of observed items for each run on each data set. The minimum number of observed items is 9 and the maximum number is one item less than total number of items.

For each dataset there exists a training set that contains 9 folds and a test set that which represents a single fold. A list of required parameters are presented in table 5.2. Samples are assigned randomly to each fold and this setting is the same across all predictive models for each run. Since all items are presented in the training set, then we can estimate the parameters that are related to items.

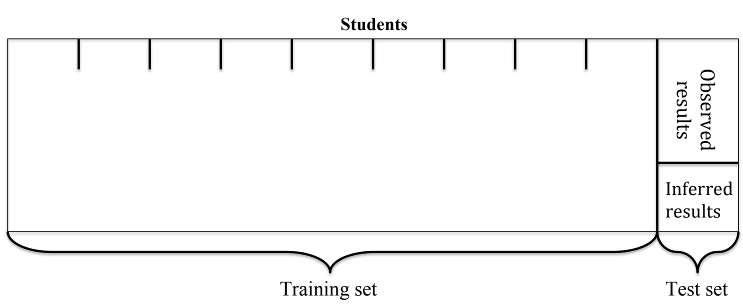


Figure 5.2 Data breakdown of cross validation process

Parameter	Typical values	Models affected
Data specific parameters		
Number of skills	3 to 9	Multiple skills models: DINA, DINO, NMF Conj./Add.
Number of items	10 to 50	All models
Number of students	100 to 1500	All models
Test success rate	0.25 to 0.85	All models
Student score variance	0.03 to 0.20	All models
Item score variance	0.03 to 0.20	All models
Item discrimination	0.5 to 3	IRT
Item difficulty	-3 to 3	IRT
Student ability	-4 to 4	IRT
Simulation parameters		
Number of observed items	9 to Number of items -1	All models
Training set size	90% of $[100-1500]$	All models
Model specific parameters		
Guess and slip	$[0.0-0.2]$	DINO and DINA
Binomial and interaction tests	$\alpha_1 = 0.85$ $\alpha_2 = 0.10$	POKS

Table 5.2 Parameters of the simulation framework

Contributed skills		Skills Model	Parameters estimated from: Training set		Parameters estimated from: Observed items
	Multiple	NMF Conj.		• Q-matrix	• Students skills mastery matrix
	Multiple	NMF Add.		• Q-matrix	• Students skills mastery matrix
	Multiple	DINA	• Slip • Guess	• Q-matrix	• Students skills mastery matrix
	Multiple	DINO	• Slip • Guess	• Q-matrix	• Students skills mastery matrix
	Single	IRT	• Item difficulty • Item discrimination		• Student Ability
	Single	Expected	• Item Odds		• Student Odds
	Zero	POKS	• Initial Odds • Odds ratio • PO structure		

Table 5.3 Parameters of the predictive performance framework

For other parameters that are related to students we need to divide the test set into an observed and inferred set. From the observed set we can get to the parameters that are related to students. A list of required parameters for assessing the model performance in table 5.3.

Once all the required parameters are presented we can make a prediction for the inferred cells of the result matrix. Note that the selected observed and inferred items are the same across all the models for each run to make a better comparison for their prediction. A probability of mastery is obtained and rounded, resulting in a 0/1 error loss function. We report the mean accuracy as the performance measure. The R package `ltm` is used for parameter and skills estimation for IRT model and the R package `CDM` and `NMF` for Deterministic noisy and NMF models. A comparison between the predicted results and the real values can result in a performance accuracy.

5.4.1 Data sets

The performance of the models is assessed over a total of 14 data sets, 7 of which are synthetic, and 7 are real data. They are listed in table 5.4, along with the number of skills of their Q-matrix, their number of items, the number of the student respondents, and the average score. Table 5.4 also reports the Q-matrix used. As can be seen, some synthetic data sets share their Q-matrix with real data sets. This sharing allows greater similarity between the synthetic data and a real data counterpart that shares a Q-matrix. Other parameters used to create the synthetic data sets were also obtained from real data sets with the same intent of allowing better comparison.

Of the 7 real data sets, only three are independent. The other 4 are variations of a well known data set in fraction algebra from Tatsuoka's work (**?**). They consists in subsets of questions and variations of the Q-matrix. These variants allows us to explore the effect of different models (Q-matrices) over the same data source.

The Vomlel data was obtained from (**?**) and is also on the topic of fraction algebra. The Q-matrix for this data is derived from the Bayesian Network defined over the 20 item test by experts.

The ECPE data (Examination for the Certificate of Proficiency in English) is an English as a foreign language examination. It is recognized in several countries as a test of advanced proficiency in English and used by a number of universities.

These real data sets were obtained from different sources and are freely available from the CDM (**?**) and NPCD (`http://cran.r-project.org/web/packages/NPCD/`) R packages. The Q-matrices of the real data sets were made by experts.

The synthetic data sets are generated from each skills assessment model, with an effort to fit the parameters as closely as possible to a real data counterpart that shares the same Q-matrix.

Data set	Number of			Mean Score	Q-matrix
	Skills	Items	Students		
Synthetic					
1. Random	7	30	700	0.75	$\mathbf{Q}_{01}$
2. POKS	7	20	500	0.50	$\mathbf{Q}_{02}$
3. IRT-2PL	5	20	600	0.50	$\mathbf{Q}_{03}$
4. DINA	7	28	500	0.31	$\mathbf{Q}_{5}$
5. DINO	7	28	500	0.69	$\mathbf{Q}_{6}$
Linear (Matrix factorization)					
6. Conj.	8	20	500	0.24	$\mathbf{Q}_{1}$
7. Comp.	8	20	500	0.57	$\mathbf{Q}_{1}$
Real					
8. Fraction	8	20	536	0.53	$\mathbf{Q}_{1}$
9. Vomlel	6	20	149	0.61	$\mathbf{Q}_{4}$
10. ECPE	3	28	2922	0.71	$\mathbf{Q}_{3}$
Fraction subsets and variants of $\mathbf{Q}_1$					
11. 1	5	15	536	0.53	$\mathbf{Q}_{10}$
12. 2/1	3	11	536	0.51	$\mathbf{Q}_{11}$
13. 2/2	5	11	536	0.51	$\mathbf{Q}_{12}$
14. 2/3	3	11	536	0.51	$\mathbf{Q}_{13}$

Table 5.4 Datasets

For POKS, the structure was obtained from the Fraction data set and the conditional probabilities were generated stochastically, but in accordance with the semantic constraints of these structures and to obtain an average success rate of 0.5.

For IRT, the student ability distributions was obtained from the Fraction data set, and the item difficulty was set to reasonable values: averaging to 1 and following a Poisson distribution that kept most values between 0.5 and 2 (done by generating random numbers from a Poisson distribution with lambda parameter set to 10 and dividing by 10).

The matrix factorization synthetic data sets of DINO and DINA were generated by taking a Q-matrix of 7 skills that contains all possible combinations of 1 and 2 skills, which gives a total of 28 combinations and therefore the same number of items. Random binary skills matrix (which corresponds to matrix $\mathbf{S}$ in equation (2.1) were generated and the same process was used for both the DINO and DINA data sets. Item outcome is then generated according to equation (2.6) with a slip and guess factor of 0.1.

A similar process was followed to generate the Q-matrices and the skills matrices $\mathbf{S}$ of the linear matrix factorization data sets, except that item outcome follows equation (2.1) and is discretized.

Note that the first 4 models do not rely on any Q-matrix for the data generation process, but the DINO/DINA and matrix factorization assessment models still require one. To define these Q-matrices (denoted $\mathbf{Q}_{0x}$ in table 5.4, a wrapper method was used to first determine the number of skills according to (**?**), then a Q-matrix was derived with the deterministic ALS algorithm as described in section 2.4.7, starting with an initial random Q-matrix.

5.4.2 Predictive performance results

This section shows the result of predictive performance of the seven models over the 14 datasets described in table5.4.

Figures 5.3 and 5.4 show the performance of each model over the synthetic and real data sets. The performance is the difference in accuracy of each model from the expected value, which serves as a baseline. Note that the y-scale of the synthetic data is double the one of the real data sets, and therefore the differences in performance for the synthetic datasets are much wider. An error bar of 1 standard deviation is reported, computed over 10 simulation runs that each run considers four different number of observation that varies between 9 to an item less than maximum number of items, provides an idea of the variability of the results. A dataset of random data is also reported for a 0.75 average success rate.

As expected, when the generative model behind the synthetic data set is the same as the skills assessment technique, the corresponding technique's performance is the best, or close to the best. And this performance is also always above the expected value performance, except for the random data set where not model can do any different than the expected value, which is what we would expect.

For the synthetic data sets, three models reach performances that are much higher than the baseline, in the range of $20-30\%$ (DINO, Linear Conjunctive, and DINA), whereas for the three other models the gain is closer to 5% (Linear additive, POKS, and IRT).

An important observation is that the pattern of relative differences of performances across techniques varies considerably and is unique to each data set: no two data sets have the same pattern of relative performance across models. The capacity of recognizing a data set's true model relies on this uniqueness characteristic.

For the real data sets, the relative performance among the techniques shows smaller discrepancies and is closer to the baseline. Although the best performers are still significantly better than the expected values for the majority of the data sets, it is surprising to see that a majority of models do worst than the baseline for a majority of data sets. ~~It does not seem always significant.~~

The results from the subsets of the Fraction data show that the pattern of the Fraction performance data set repeats over Fraction-1, Fraction-2/1 and Fraction-2/2, in spite of the different number of skills and different subsets of questions. However, it differs substantially from Fraction-2/3 for

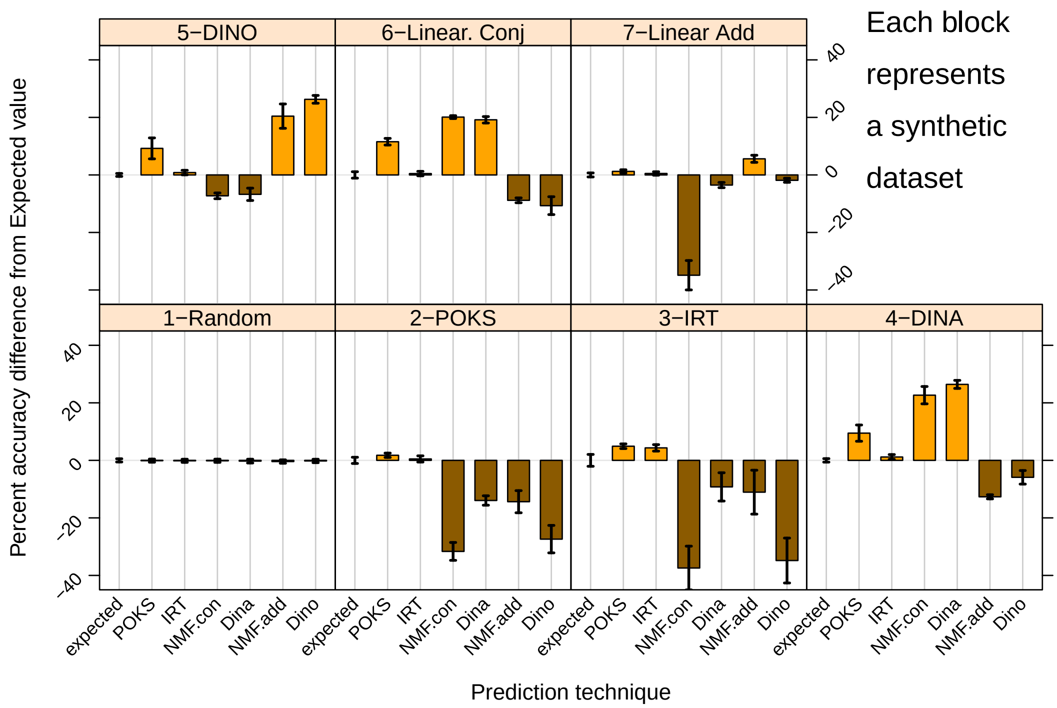


Figure 5.3 Item outcome prediction accuracy results of **synthetic data sets**

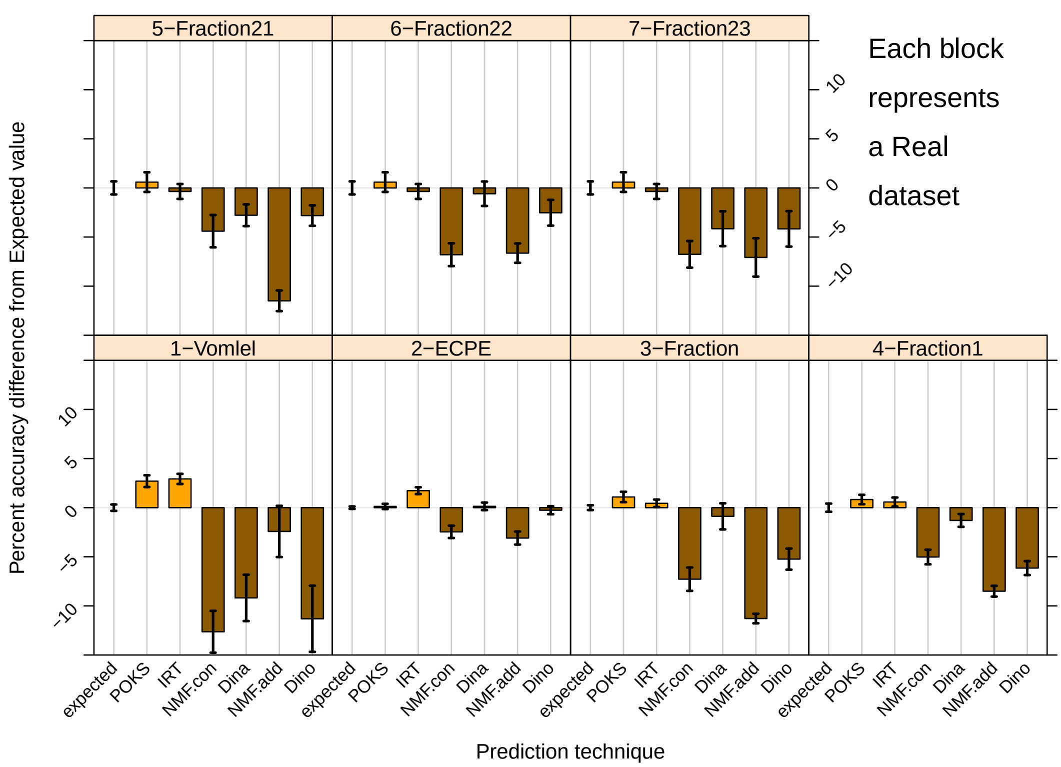


Figure 5.4 Item outcome prediction accuracy results of **real data sets**

the NMF conjunctive performance which reaches that of the NMF additive one and also DINA reaches DINO. This is readily explained explained by the fact that the Q-matrix of this data set has the property of assigning a single skill to each item, in which case the two matrix factorization techniques become equivalent. But aside from the Fraction-2/3 case, this similarity among Fraction data set and its derivative suggests that in spite of the model differences (different Q-matrices and item subsets), the performance "signature" remains constant across these data sets.

Finally, we note that none of the real data sets show the large the discrepancies found in the synthetic data sets models. One exception is the linear additive synthetic data set which displays smaller variance across models and which "signature" resembles the Vomlel data, although the performance difference with the majority class is substantially higher for the synthetic data than the Vomlel data, suggesting that the real data is yet not a perfect fit to this model.

5.5 Experiment 2: Sensitivity of the Model performance over data generation parameters

In this experiment we want to examine the effect of data generation parameters on the stability of the model performance vector in the performance space. In this section we run the same experiment but with different parameters such as average success rate, sample size, number of latent skills, number of items, student and item score variance. These factors can can answer the question whether the patterns hold across different conditions. Table 5.2 shows different conditions of the parameters.

Just like the previous experiment, we assessed these results on the basis of 10-folds cross-validation. The same as before, we fixed the number of "observed items" for each run on each data set that varies between 9 to one item less than total number of items.

We generated the synthetic datasets with different sample size which varies from 100 students upto 1500 students. Figure 5.5 shows the result of this change. Running all these techniques on the synthetic datasets did not change any pattern except for IRT generated dataset that slightly changed the pattern for small sample size. On the other cases the one with the highest predictive performance was the same as model behind the generated data. Also changing the sample size downto less than 100 students will change the pattern as well; because the training set of the model could not be Learned properly.

The other parameter is the number of Items in generation of the synthetic dataset. This time we fixed all the other parameter but the Items size which varies between 10 to 50 items. Figure 5.6 shows how signature has been changed for different models. The results on the synthetic datasets shows that in some cases the pattern stays the same but it shifts down on the chart once we decrease the number of items. Still we can see that the highest performance is the same model behind the generated dataset. This highlights the role of this variable in predicting the signature and also

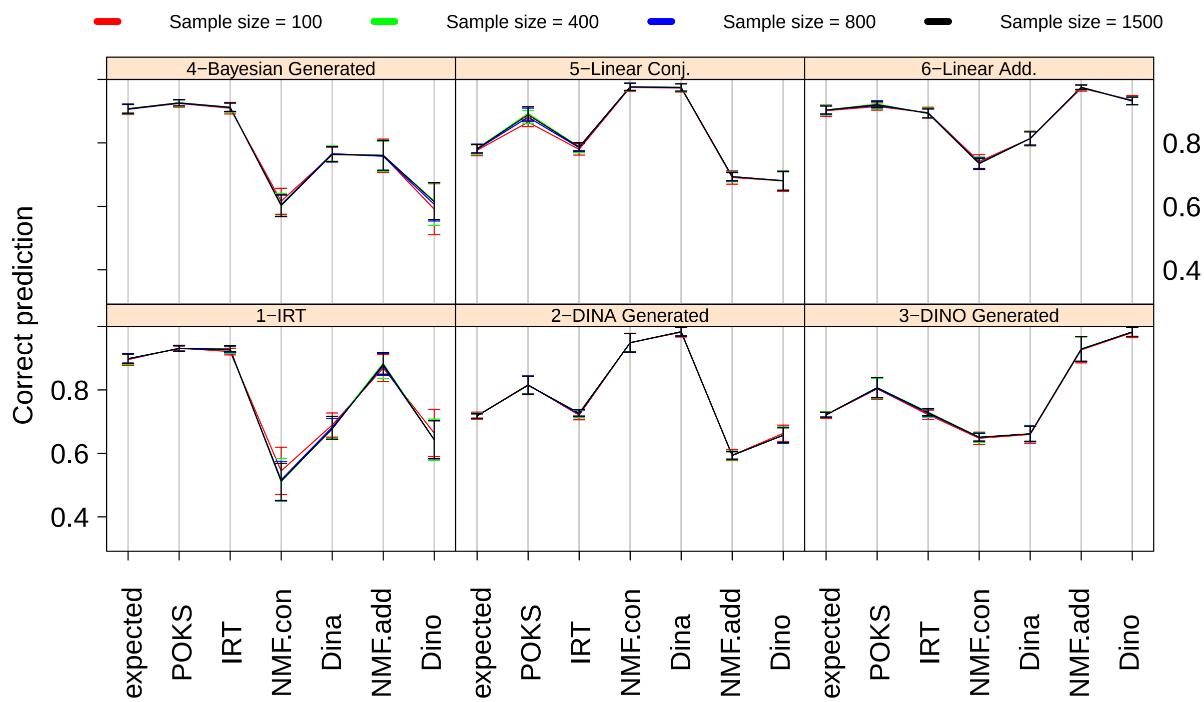


Figure 5.5 Variation of **Sample Size** Over synthetic data sets

means that the highest performance should not necessary be close to 100%.

Next variable is number of skills in generation of dataset and estimation of the signature. Same as before we fixed all the other variables to generate the dataset but the number of skills behind the generation of the synthetic dataset which has been in the range of 3 to 9 skills in total. Figure 5.7 shows the result of this experiment. For IRT and POKS generated dataset it does not make sense to have number of skills because IRT is a single skill model and there are no explicit latent skills. In order to find the predictive performance of linear methods for bayesian or IRT generated datasets we need to have a fixed number of skills as a part of prediction and that is how we run the experiment with different number of latent skills. The signature pattern did not change substantially for these datasets. Only for linear based techniques we can find a slight change in the pattern. For linear models the pattern still stays the same but for NMF additive when the number of skills increases the ground truth is not the highest performance in the signature. Increasing the number of skills while the number of item is constant can resemble the single skill modeling such as IRT and POKS.

Figure 5.8 shows the same experiment across datasets with different item variance. The item variance was reflected as the item difficulty for IRT and initial odds for POKS. The value for item variance shows the standard divination of the normal distribution of the items. Clearly the pattern stays the same for different values except for few minor changes. Although the the set of

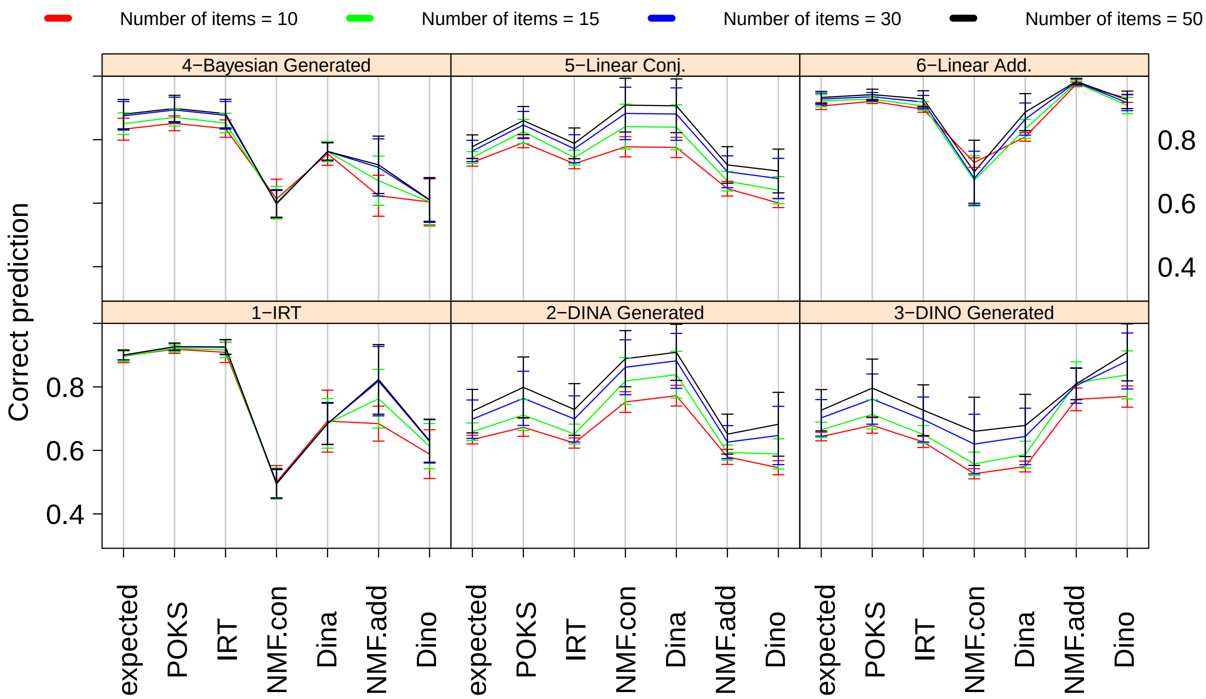


Figure 5.6 Variation of **Number of items** Over synthetic data sets

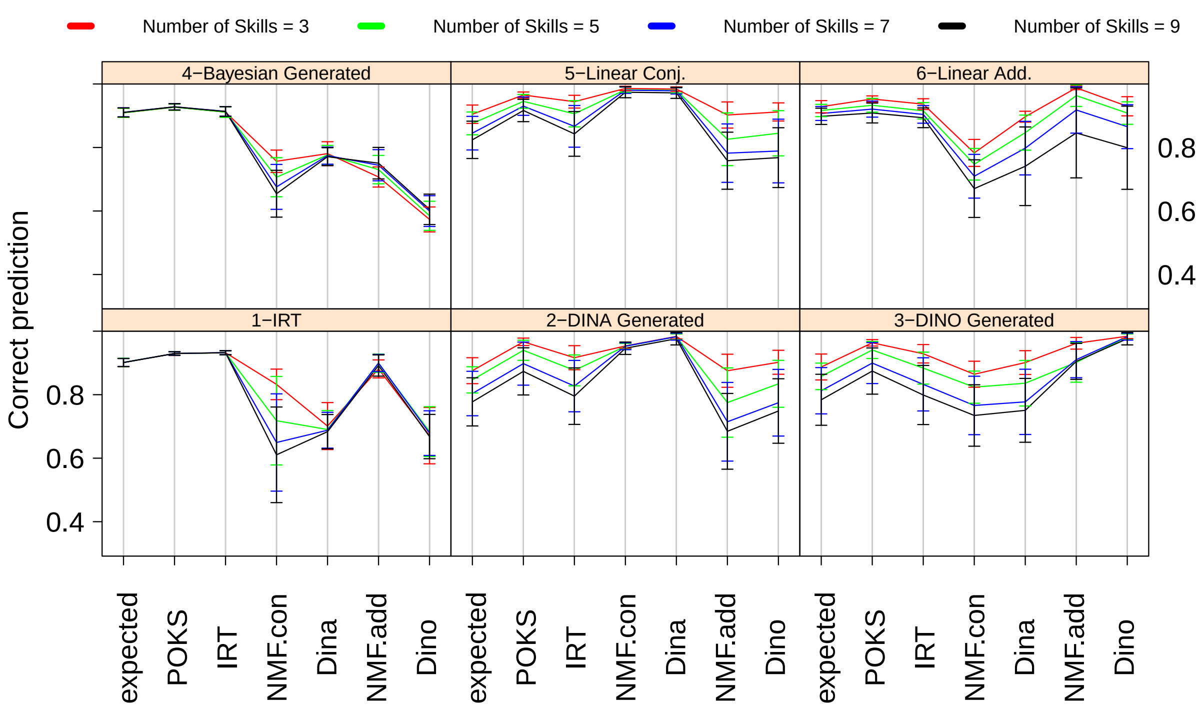


Figure 5.7 Variation of **Number of skills** Over synthetic data sets

performance values shifts across correct prediction axis but the pattern stays the same.

Finally, we did a last experiment on datasets with different success rates. For synthetic data, the success rate starts from 0.25% up to 0.85%. The pattern is scaled for some linear methods on IRT and bayesian generated data and it doesn't change the performance of IRT and POKS themselves. For datasets with linear structure the signature was shifted downward on linear methods predictive performance side as we decrease the success rate for data generation.

5.5.1 Degree of similarity

The most important part is to have a measure to show the degree of similarity between the model performance vector of the synthetic data and the one of the ground truth which is a measure of model fit in our study. Many measures can be used to calculate this degree. The simplest one can be finding the distance between two vector of predictive performance between the synthetic dataset and the real one. previous section showed that different data generation parameters with the same underlying model can create slightly different model performance vectors but with the same pattern. Using distance between each of these signatures with the same ground truth can possibly result in different values for similarity. In this research we used Pearson correlation coefficient as a measure of similarity.

The evaluation of this research consists of two parts. The first part is to measure the degree of similarity for synthetic vs. synthetic datasets. The second part is to identify the ground truth by measuring the correlation between real and synthetic datasets.

Given 6×4 different set of data generation parameters (described in section 5.5) and generating 10 times for each model will produce 1440 datasets. The comparison in our study is based on same data specific parameters. Therefore we can calculate the correlation table among 60 synthetic datasets that share same data specific parameters. This will result in 24 tables which consider all the possible data generation parameters. The expectation is that those vectors with same model show high correlation. Table 5.5 shows the average correlation of these 24 conditions.

Evaluating the synthetic dataset versus themselves can give a clue to how accurate this comparison is in model fitting. Basically, Those datasets that have the same model behind them should show a good correlation. The diagonal of table 5.5 shows a high correlation because it compares the same model generated datasets. On the other hand some models such as IRT and POKS shows a high correlation since they are not using multi-skills models. Those models that share concepts such as DINA and NMF conjunctive also resulted in a high correlation comparing with other models because they are linear models which deal with conjunctive model of Q-matrix. DINO and NMF additive has almost a high correlation but since the additive model is slightly different from the

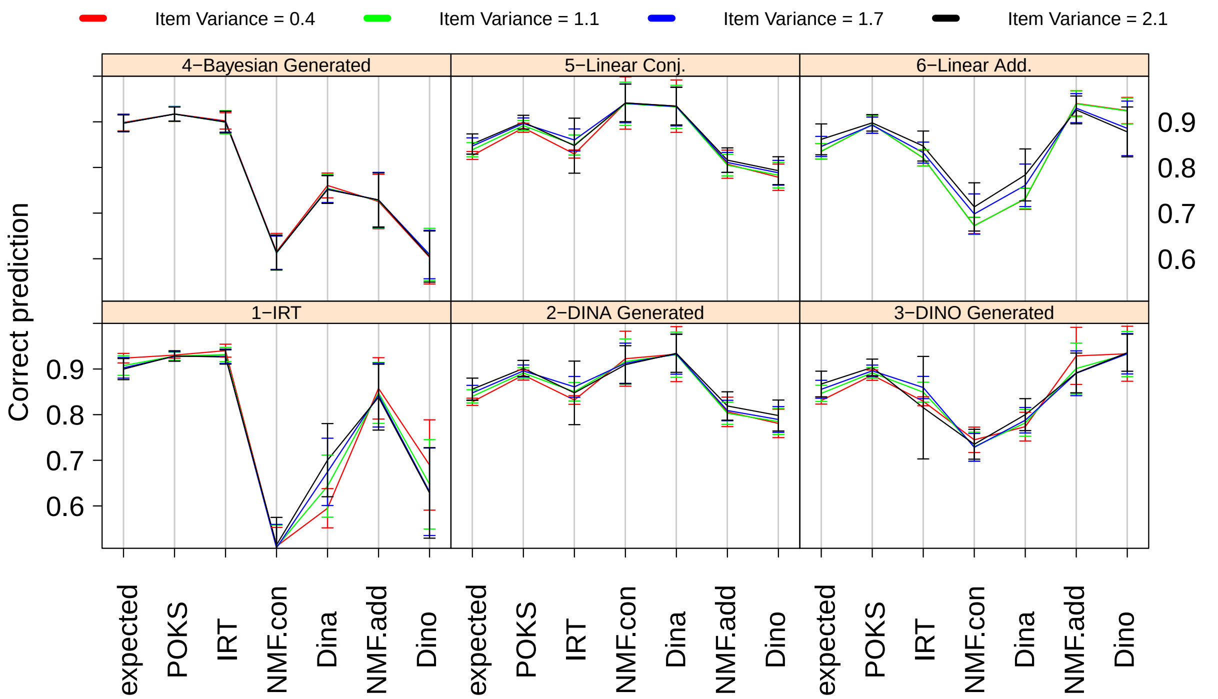


Figure 5.8 Variation of **Item Variance** Over synthetic data sets

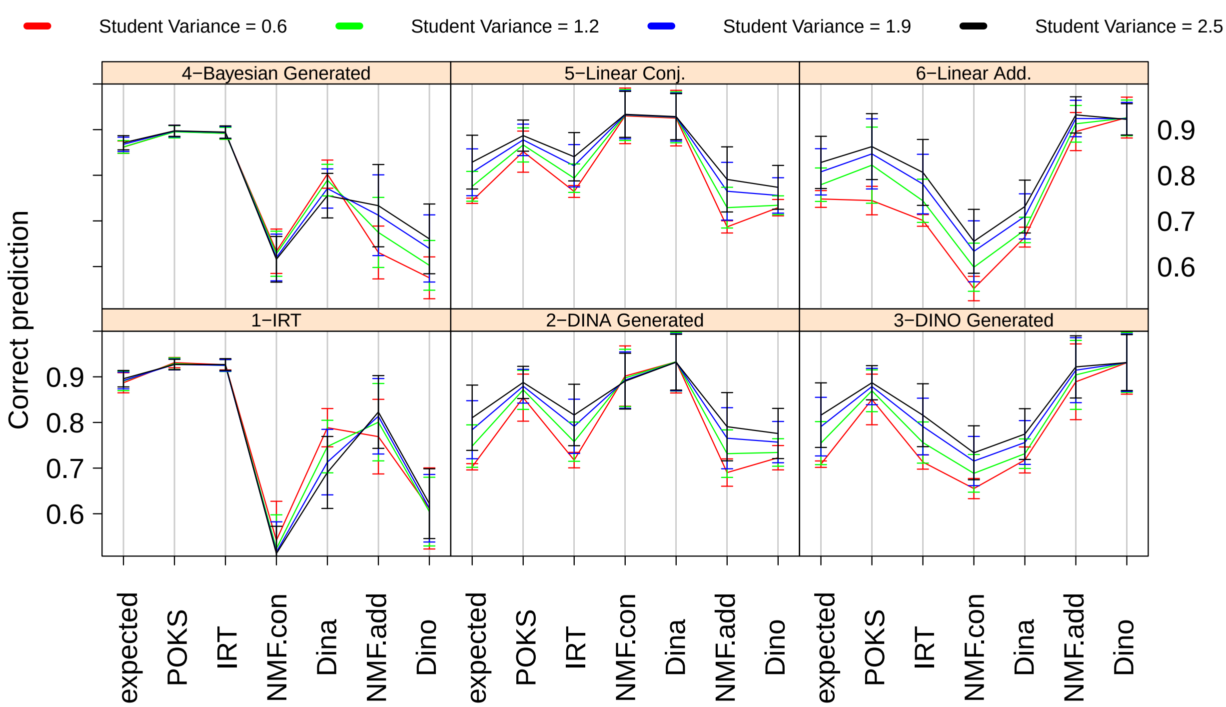


Figure 5.9 Variation of **Student variance** Over synthetic data sets

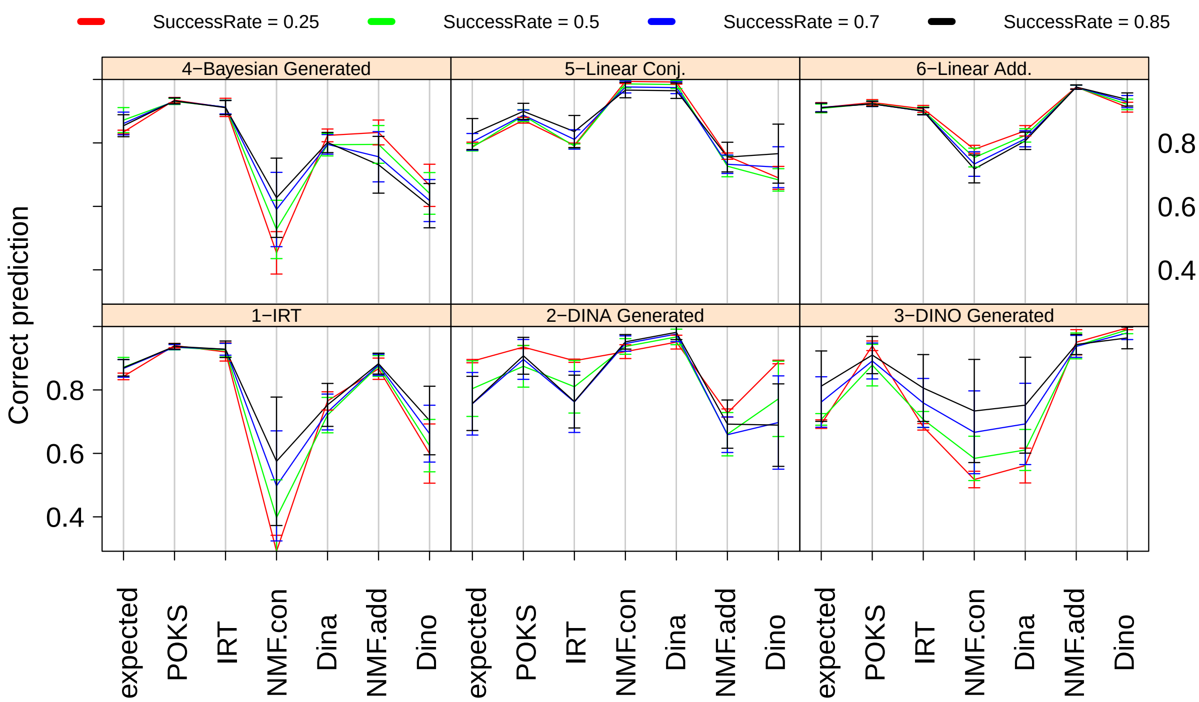


Figure 5.10 Variation of **Success Rate** Over synthetic data sets

disjunctive model then it's less correlated.

To assess a model fit for each real dataset with this approach we need to create datasets with described underlying models (7 techniques) which have the same parameters of data generation as the real data. Since some parameters influence the performance vector as described in the section 5.5 so we have to create synthetic data that follows the same characteristics of the real one. Table 5.6 shows the correlation of real data signature in columns with the signature of synthetic data generated with underlying model in rows. Vomlel dataset shows a high correlation with IRT model and

Synthetic Datasets

Synthetic Datasets	POKS	IRT	NMF Conj.	DINA	NMF Add.	DINO
POKS	**0.96**					
IRT	0.86	**0.96**				
NMF Conj.	0.22	-0.20	**0.96**			
DINA	0.02	-0.40	0.94	**0.96**		
NMF Add.	0.44	0.75	-0.62	-0.73	**0.93**	
DINO	-0.15	0.20	-0.70	-0.69	0.63	**0.95**

Table 5.5 Degree of similarity between six synthetic datasets based on the correlation

	Real Datasets						
				Fraction subsets			
Synthetic Datasets	Vomlel	ECPE	Fraction	1	21	22	23
Random	0.58	**0.73**	0.61	0.43	0.24	0.61	0.57
IRT	**0.90**	0.42	0.72	0.88	0.60	0.77	0.61
DINA	-0.38	-0.09	0.23	0.30	0.56	0.06	0.38
DINO	0.34	0.15	-0.18	-0.31	0.10	-0.08	0.38
POKS	0.75	0.40	**0.83**	**0.95**	**0.70**	**0.83**	**0.80**
NMF Conj.	-0.05	0.54	0.51	0.55	0.66	0.33	0.57
NMF Add.	0.39	0.06	-0.04	-0.19	-0.03	0.13	0.28

Table 5.6 Degree of similarity between six synthetic datasets and the ground truth based on the correlation

Fraction with its subset datasets show similarity with POKS model. As expected ECPE has the highest correlation with random generated dataset.

The very last experiment that we have done is a comparison between the proposed approach where the nearest neighbor classifies in the performance vector space and the simplest approach where the best performer is defining the ground truth. The process is straightforward which is preforming a classification for all synthetic data with best performer method. For the nearest neighbor approach the classification should be done where datasets have the same data generation conditions. Therefore we'll get 24 sets of datasets where similar dara generation parameters exists among each set.Since we have 10 runs for each data specific parameter set then we choose 10 nearest neighbor and preform majority voting on the classification results.

Table 5.7 shows the confusion matrix of this experiment. There exists 1440 datasets where each

BP: Best Performer
NN: Nearest Neighbor

	Datasets													
	POKS		IRT		NMF Conj.		DINA		NMF Add		DINO		Accuracy (%)	
Models	BP	NN	BP	NN	BP	NN	BP	NN	BP	NN	BP	NN	BP	NN
Expected	0	0	0	0	0	0	0	0	12	0	2	0		
POKS	238	218	130	32	21	12	14	0	18	13	1	0	85	95
IRT	2	20	110	208	0	0	0	0	0	15	0	3	100	97
NMF Conj.	0	0	0	0	82	180	5	73	0	0	0	0	100	94
DINA	0	0	0	0	137	48	221	167	0	0	0	0	87	96
NMF Add.	0	2	0	0	0	0	0	0	210	211	14	10	99	99
DINO	0	0	0	0	0	0	0	0	0	1	223	227	100	100
Accuracy (%)	99	91	46	87	34	75	92	70	88	87	93	95		

Table 5.7 Confusion matrix for classification of 210 synthetic datasets on 7 models with Best performer Vs. Nearest neighbor methods

	Performance							
	Best Performer				Nearest Neighbor			
Models	Precision	Recall	F-Measure	Accuracy	Precision	Recall	F-Measure	Accuracy
POKS	0.564	0.992	0.719	0.871	0.793	0.908	0.847	0.945
IRT	0.982	0.458	0.625	0.908	0.846	0.867	0.856	0.951
NMF Conj.	0.943	0.342	0.502	0.887	0.711	0.750	0.730	0.907
DINA	0.617	0.921	0.739	0.891	0.777	0.696	0.734	0.916
NMF Add.	0.938	0.875	0.905	0.969	0.946	0.879	0.911	0.971
DINO	1	0.929	0.963	0.988	0.996	0.946	0.970	0.990

Table 5.8 Accuracy of best performer and nearest neighbor classification methods

model corresponds to 240 dataset. The gray cells in table 5.7 shows the true positive values and other values in each column represent the false positive predictions for group of datasets.The values in each row shows the number of false negative predictions for each model. The confusion is mostly between those techniques that shares same concepts specially between NMF Conjunctive and DINA model where we use conjunctive Q-matrices.

The accuracy that is reported in the last row of table 5.7 is calculated based on $\frac{TP}{240}$ which counts the true positive predictions for each sub set of datasets that have the same model behind them. The accuracy the is reported in the last two columns of table 5.7 is considering how faithful the classification is to select a dataset that doesn't have the related model behind it which will count true negative values based on $\frac{TN}{1200}$ (1200 is the number of datasets that do not have same underlaying models). In terms of true positive selections there is no benefit between any of these methods even sometimes best performer shows to be better (specially for DINA and IRT).

Considering the false negative and false positive changes the classification results. Table 5.8 shows the accuracy of this classification in terms of precision, recall, $F1$ measure and accuracy ($\frac{TN+TP}{1440}$) . Since $F1$ measure is combining both precision and recall, then it is a good measure for improvement. The third column of each classification method shows that F-measure increased for Nearest neighbor method which is almost close to 1. Also in terms of individual scores per method we also report accuracy of each technique which considers true positive and true negative values. The last column of table 5.8 shows this improvement. The total accuracy which is considering true positive numbers over number of datasets regardless of individual models shows that best performer gets 0.75% and the nearest neighbor gets upto 0.84% of accuracy.

CHAPTER 6
Conclusion and future work

6.1 Conclusion

In this thesis, we tackled few contributions that are applying some techniques on Q-matrices and assessing model fit with synthetic vs. real data which is the main contribution of this research . In this chapter, we summarize the results, conclusions, and possible future works.

Let us return to the conjecture that the comparison of real vs. synthetic data can help determine whether a specific skill model corresponds to the ground truth of some data set. This is a complex question but some hints are given in the results.

A clear finding is that the synthetic data sets have very distinct performance patterns, showing sharp differences across models. In that respect, synthetic data do have a distinct patterns or specifically distinct positions in performance space. And none of the real data sets display the sharp differences found in all but one of the data sets, namely the data generated from the NMF additive model.

The Fraction data sets do display a similar pattern of performances across different subsets of items, different number of skills (latent factors), and different variants of the models as expressed by variations in the Q-matrices. Only when the Q-matrix has the property of a single skill per item do we observe a very different performance vector for the models that depend on the Q-matrix (NMF conjunctive, DINA, NMF additive, and DINO). The other models are not affected (expected, POKS and IRT). Therefore, in the domain of skills modeling, we find evidence that data from a common source does have correlated performance vectors as long as the models do not have large formal differences.

6.2 Future Work