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. GUHNEUC Yann-Gal, Doctorat, président
M. DESMARAIS Michel C., Ph. D., membre et directeur de recherche
M. GAGNON Michel, 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

LIST OF TABLES

LIST OF FIGURES

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

In Educational Data Mining, or in Data Science in general, 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. Selection is most often based on a model's "goodness of fit".

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.

Another approach to assess which model is closest to the ground truth is proposed in this thesis. It relies on the analysis of model predictive performances over real data and 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 Thesis vocabulary

In this section we introduce a vocabulary that is related to the general objective of this thesis and used in all chapters:

— Dataset (Real/Synthetic): Dataset in this context represents student test outcome which is a matrix that shows the result of a test given by students. A test is simply a set of

few questions, problems or items that can have a success of a failure result in the dataset. Datasets can be “real” or “synthetic”. A “Real” dataset is the result of an actual test given by individuals in an e-learning environment or even a classroom. The term “Synthetic” means that a simulation is involved to generate an artificial student test outcome. The simulation is designed based on a model that takes a set of predefined parameters to generate student test outcome. This set contains two types of parameters : Model specific parameters and Data specific parameters.

— Model specific parameters: These parameters are specifically defined and learnt based on model’s type. Complex models contain more parameters. Some models may share some parameters but some models have no parameters in common.

— Data specific parameters: These parameters are common between all datasets such as average success rate, sample size and number of items in a dataset.

— Ground truth: This term is originally coined by Geographical/earth science where if a location method such as GPS estimates a location coordinates of a spot on earth, then the actual location on earth would be the “Ground truth”. This term has been adopted in other fields of study. In this context “ground truth of a dataset” means the actual model that best describes the dataset within its parameters.

— Performance of a model: The accuracy of a model over a dataset is called performance of a model. This accuracy is obtained based on the correct prediction of student test outcome with that model. Different models have different performances over a dataset. Assessing such a performance requires designing an experiment to learn the model’s parameters and predict a proportion of the dataset that has not been involved in the learning phase.

— Performance vector: Each model have a performance over a dataset and these performances can be different values of accuracy. This results in a large space in which to explore model performance. In this thesis we defined a vector space based on model performances in which a dataset can have a performance vector that combines all performances of candidate models by putting them together in a vector.

— Signature: The performance vector can be considered from two perspectives: The first perspective is a performance vector in the performance space where we have the same number of dimensions as the number of candidate models in the performance vector. The second one is a kind of “signature” for a specific data which considers the vector in a two dimensional space with performances on the y axis and skills assessment models on x axis. They are sharing the same concepts but different presentations.

1.3 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?

1.4 General Objectives

The general objective of this thesis is to assess the goodness of fit on the basis of the what we will refer to as performance signatures. It can be divided in three sub-objectives: The first objective is to obtain the performance signatures of skills assessment models over a synthetic datasets generated with these very 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.

1.5 Hypotheses

The research in this thesis tests the following hypotheses:

Hypothesis 1: The signatures 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. [Behzad says : or Relative performance of each synthetic data type is unique]

[Behzad says :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 ground truth but different data specific parameters can have different performance vectors.

1.6 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 focus on student test result and on a 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.

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 estimate 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 a 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 the "signature" to 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 over different data specific parameters (such as sample size, average success rate, etc.) generated with the same underlying model.

1.7 Publications

Along the course of the doctorate studies, I contributed to a number of publications, some of which are directly related to this thesis, and some of which are peripheral or are preliminary studies that led to the thesis.

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.
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.

1.8 Organization Of the Thesis

We review the 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 recent work about model selection. We explain synthetic data generation approaches in

chapter 4. The main contribution of the research is explained in details in Chapter 5. Finally, we conclude and outline future work in Chapter 6.

CHAPTER 2
STUDENT MODELLING METHODS

A large body of methods have been developed for student modeling. They are used to represent and assess student skills. These models have been proposed both for dynamic performance data, where a time dimension is involved and where a learning process occurs (see for eg. Bayesian Knowledge tRacing **?**), and for static performance data where we assume student skill mastery is stationary. In this thesis, we focus on static performance data.

We assume that skills explain the performance and outcome prediction. In general there exists many student models that incorporate zero, single or multiple skills. The most widely used one is Item Response Theory (IRT). In its original and simplest version, IRT considers that a single skill student performance data. Of course, sometimes there are many skills involve in a single problem and therefore this model becomes insufficient for many applications in Intelligent Tutoring Systems (ITS). Under certain condtions, multi-skills models can preform better in that case. Other methods, such as POKS, have no latent skills. They just look at the relation between what directly observable test outcome items. The details of each category of techniques along with some examples are described in the next section.

2.1 Definitions and concepts

In this section some concepts and basic definitions that are common between most of student models are described.

2.1.1 Test outcome data

The student test outcome data, or more simply student test 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. If a student successfully answers an item, the corresponding value in the results matrix is 1, otherwise it is 0.

2.1.2 Skills

In this thesis we consider skills as problem solving abilities. For example in mathematics "addition", "division" are typical skills. They can also be further detailed, such as single digit and multi-digit addition. There may be a single skill required to solve a problem or multiple skills. Skills are termed *latent* because they are never observed directly.

If an item requires multiple skills, there are three different combinations in which each skill can contribute to succeed a problem: The first case is when having a skill is mandatory for a student to succeed an item that requires that skill. The second case is when the skill increases the chance to succeed a problem. The third case is when at least one of the skills from the set of skills for an item is required to succeed that item. We will see later that the temrs *conjunctive*, *compensatory/additive*, and *disjunctive* are often used to refer to each case respectively.

Skills can have different range of values based on the student model definition. For example, the single skill in IRT is continous in $\mathbb{R}$ and typically ranges between -4 to $+4$. Other student models consider skills range between 0 and 1. Finally, other models have binary 1 or 0 skills, which means it can be mastered or not. Details of definition of skills for each student model are given in next sections.

2.1.3 Q-matrix and Skill mastery matrices

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 skills to items (**??**). Figure 2.1 shows an example of this mapping which is named Q-matrix. Figure 2.1(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 2.1(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. Barnes, Bitzer, & Vouk in (**?**) were among the early researchers to propose algorithms for automatically discovering a Q-Matrix from data. In some context there is a constraint for the number of latent skills which is: $k < nm/(n+m)$ (**?**)

where k , n and m are number of skills, students and items respectively.

i_1 $\quad \frac{4}{\frac{12}{3}} + \frac{3}{5} = \frac{8}{5}$

i_2 $\quad \frac{4}{\frac{12}{3}} = \frac{4\times 3}{12} = \frac{12}{12} = 1$

i_3 $\quad 1 + \frac{3}{5} = \frac{8}{5}$

i_4 $\quad 2 \times \frac{1}{2} = 1$

(a) Items with different skills

Items	Skills s_1	s_2	s_3
i_1	1	1	1
i_2	1	0	1
i_3	0	1	1
i_4	1	0	1

(b) Q-matrix

Figure 2.1 Items and Q-matrix

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.

2.1.4 Types of Q-matrices

As explained earlier, skills can have three interpretation when they contribute to succeed an item. These interpretations can be reflected in Q-matrices. There exists three different types of Q-matrices which are useful based on the context of the problem domain:

— **Conjunctive**: The most common one is the conjunctive model of the Q-matrix which is the standard interpretation of the Q-matrix in Educational Data Mining. In this Q-matrix type, a student should master all the required skills by an item to succeed it. If a student misses any one of these skills, then the result will be a failure in the test outcome data. Thus there is a conjunction between required skills to succeed an item.

— **Additive** (compensatory): Compensatory or additive model of skills is an interpretation of a Q-matrix where each skill contributes some weight the 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 skill increases the chance of success based on its weight.

— **Disjunctive**: In the disjunctive model, mastery of any single skill required by an item is sufficient in order to succeed the related item.

Q-matrices can also be categorized according to the number of skills per item:

— Single skill per item: Each item should have exactly one skill but the Q-matrix can have many skills.

— Multiple skills per item: Any combination of skills with at least one skill is possible for items.

Note that the three types of interpretation of Q-matrix make sense when there is at least one item in the Q-matrix that requires more than one skill. In the next sections of this chapter these types will be described in more details with examples.

2.2 Skills assessment and item outcome prediction techniques

The skills assessment models we compare can be grouped into four categories: (1) the Knowledge Space frameworks which models a knowledge state as a set of observable items without explicit reference to latent skills, (2) the single skill Item Response Theory (IRT) approach, (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 DINA/DINO approaches which also refer to a Q-matrix, but incorporate slip and guess factors and rely on different parameter estimation techniques than the matrix factorization method. 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 2.2 shows this hierarchy of models.

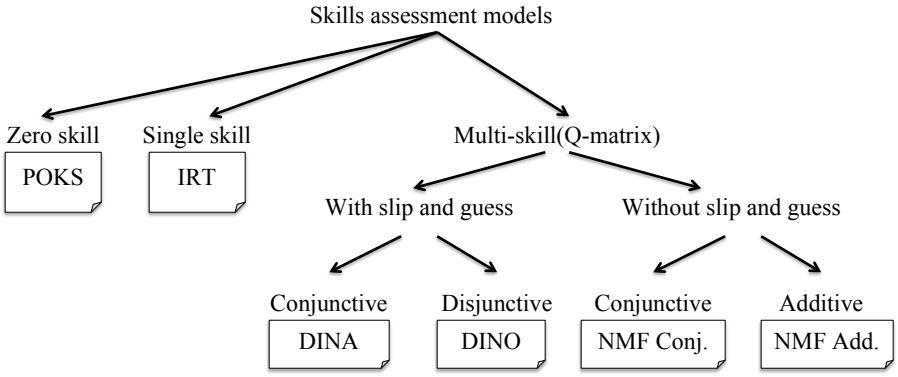


Figure 2.2 Skills assessment methods

Considering these techniques from the perspective of variety of skills in test outcome prediction,

we can put them in the following categories:

— Zero skill technique that predict item outcome based on observed items. POKS is the technique that is used as a zero skill student model.
— Single skill approaches, where every item is linked to the same single skill. Item Response Theory (IRT) is the typical representative of this approach, but we also use the "expected value" approach which, akin to IRT, incorporates estimates of a the student's general skill and the item difficulty to yield a predicted item outcome.
— Multi-Skills techniques that rely on Q-matrices to predict test outcome. Deterministic Input Noisy And/Or (DINA/DINO), NMF Conjunctive and NMF additive are the techniques we use in this study.

Note that the "expected value" approach is considered as a baseline for our evaluations.

The details of the different approaches are described below.

2.3 Zero skill techniques

Zero skill techniques are so called because they make no explicit reference to latent skills. They 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. An individual's knowledge state is represented as a subset of these items. In place of representing skills mastery directly, they leave the skills assessment to be based on the set of observed and predicted item outcomes which can be done in a subsequent phase.

POKS is one of the models adopted in our study that is a derivative of Knowlege Space Theory. POKS stands for Partial Order Knowledge Structures. It 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.

2.3.1 Knowledge Spaces and Partial Order Knowledge Structures (POKS)

Items are generally learnt in a given order. Children learn easy problems first, then harder problems. It reflects the order of learning a set of items in the same problem domain. POKS learns the order structure from test data in a probabilistic framework. For example in figure 5.3 four items are shown in a partial order of knowledge structure. It is required for an examinee to be able to answer i_4 in order to solve i_3 and i_2. Also for solving i_1, one should be able to answer i_2, i_3 and i_4. if an examinee was not able to answer i_4 then he would have less chance to answer correctly other items.

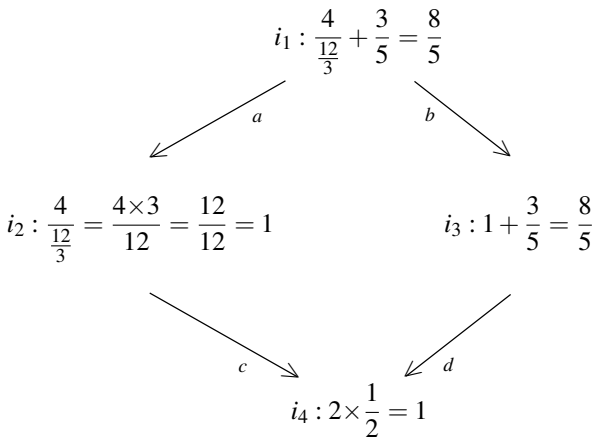


Figure 2.3 Partial Order Structure of 4 items

This is reflected in the results matrix **R** by closure constraints on the possible knowledge states. 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). The algorithm to derive such structures from the data in **R** relies on statistical tests (**??**).

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 2.4 shows the corresponded oriented incidence matrix and adjacency matrix of the structure in figure 5.3.

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 of an interaction test (α_2). The χ^2 test is often used, or the Fisher exact test. 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.

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

O A

Figure 2.4 Oriented incidence matrix and Adjacency matrix

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 :

$$O(H|E_1,E_2,...,E_n) = O(H)\prod_i^n \frac{P(E_i|H)}{P(E_i|\overline{H})} \tag{2.1}$$

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.

2.4 Single skill approaches

Other approaches incorporate a single latent skill in the model. This is obviously a strong simplification of the reality of skilled performance, but in practice it is a valid approximation as results show. When a model uses single latent skill, in fact it projects all the skills mastery level in a form of unidimentional representation that implicitly combines all skills. Then there would be a single continuous skill variable which is a weighted average of the skills mastery levels of an examinee.

In this section two approaches for modeling static test data are presented: The well established Item Response Theory (IRT) which models the relationship between observation and skill variable based on a logistic regression framework. It dates back to the 1960's and is still one of the prevailing approaches (**?**). The second approach is a trivial approach we call Expected Prediction. This approach is used as a baseline in our experiments.

2.4.1 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, where parameters are:

- a represents the item discrimination;
- b represents the item difficulty, and
- θ_i the ability of a single student.

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 use the more general IRT-2pl model, which includes both a and b, for the synthetic data generation process and test outcome prediction in order to make this data more realistic (next chapters discuss data generation in details).

2.4.2 Baseline Expected Prediction

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: $P(X) = (f(x=1) + 1)/(f(x=1) + f(x=0) + 2)$

2.5 Multi-skills techniques

Finally the last category of student skills assessment models are considering student test result with multiple latent skills. Representation of multiple skills is possible in the form of a Q-matrix where skills are mapped to each item. As explained before there exists different types of Q-matrices that

each type represent a unique interpretation. The following sections will describe different skills assessment models along with their specific type of Q-matrix.

Still there exists two types of methods in this category:

— Models that infer a Q-matrix from test result data such as models that uses matrix factorization techniques.

— Models that require a predefined Q-matrix to predict test outcome. These techniques can not directly infer the Q-matrix but they can refine an existing expert defined Q-matrix.

Deriving a Q-matrix from a test result matrix was always challenging. Some models require a pre-defined Q-matrix. In some cases an expert defined Q-matrix is available but minor mistakes in mapping skills to items are always possible even by an expert. The basic challenge is to derive a perfect Q-matrix out of a test result matrix to give it as an input parameter to some models. This challenge also creates other challenges such as optimum number of latent skills for a set of items in a test outcome. Sometimes there exists more that single Q-matrix associated with a test result with different number of skills. Finding the optimum number of skills to derive a Q-matrix is a question that is given in more details in section 2.6.2.

Given the number of latent skills, there exists few techniques to derive a Q-matrix for models that require one. 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 our experiment we use NMF as a technique to derive a Q-matrix given an optimum number of latent skills. Later in this section we will introduce NMF in more details. Section 2.6.1 describes how to derive a conjunctive model of Q-matrix from a student test result.

For real datasets there exists few expert defined Q-matrices. To use them as an input parameter of student skills assessment models we need to refine them. Section 2.6.3 gives few approaches to this problem.

2.5.1 Types of Q-matrix (examples)

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 5.2 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.

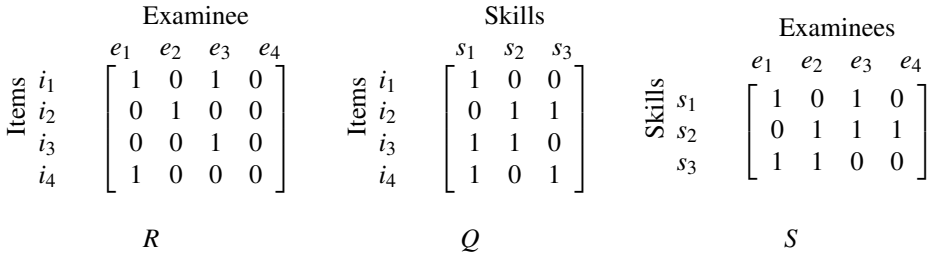


Figure 2.5 An example for Conjunctive model of Q-matrix

The other type is the additive model of Q-matrix. 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 its weight. It is possible to have different weights for skills where skills for each item will sum up to 1. Figure 2.6 represents an example of an additive model of Q-matrix with its corresponding result matrix. The value R_{ij} can be considered as a probability that examinee i can succeed item j.

Finally, in the disjunctive model of Q-matrix, there is a disjunction between skills of an item to achieve a success. At least one of the required skills should be mastered in order to succeed that item. Figure 2.7 shows an example of this type. For example examinee e_3 has both skills S_1 and S_2 and all items require either S_1 or S_2; then he should be able to answer all items in the test outcome.

Comparing these three types together, in the same condition for student skills mastery level we can find out that the average success rate in the test result data is the highest for disjunctive type and the lowest for conjunctive type. Also if we consider the Q-matrix as a single skill per item, then the definition of conjunctive, disjunctive and additive model of Q-matrix does not affect the final result.

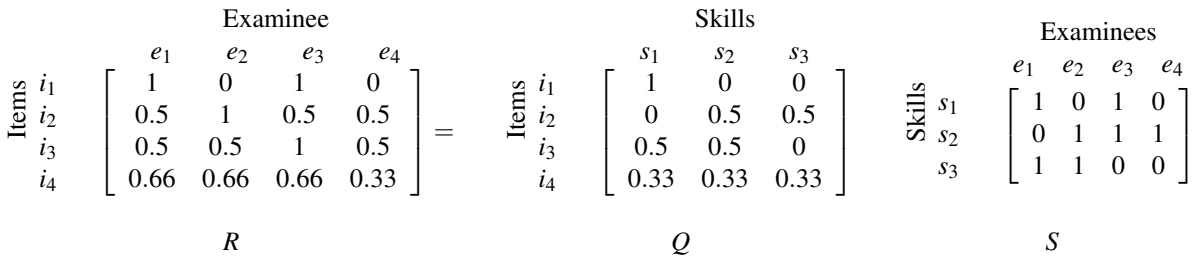


Figure 2.6 An example for Additive model of Q-matrix

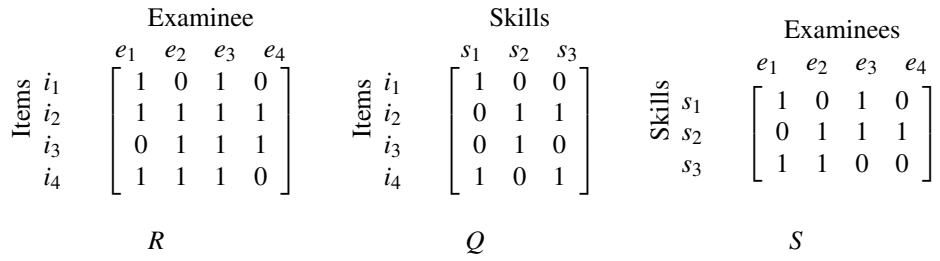


Figure 2.7 An example for Disjunctive model of Q-matrix

2.5.2 Non-Negative Matrix Factorization (NMF)

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. Singular Value Decomposition (SVD) and NMF are well known examples of such methods. Beyond skill modelling, it is an important technique in different fields such as bioinformatics, and vision, to name but a few. It has achieved great results in each of these fields. For skills assessment, using tensor factorization, a generalization of matrix factorization to a hypercube instead of matrix and where one dimension represents the time, Thai-Nghe 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. In this research, we use NMF as a skill assessment method that infers a Q-matrix with multiple skills.

Assume $\mathbf{R}$ is a result matrix containing student test results of n items (questions or tests) and m students. NMF decompose the non-negative $\mathbf{R}$, as the product of two non-negative matrices as shown in equation(2.2):

$$\mathbf{R} \approx \mathbf{QS} \tag{2.2}$$

where $\mathbf{Q}$ and $\mathbf{S}$ are $n \times k$ and $k \times m$ respectively. $\mathbf{Q}$ represents a Q-matrix which maps items to skills and $\mathbf{S}$ represents 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. Equation 2.2 represents an additive type of Q-matrix.

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 matrix as well. In this example the product of $\mathbf{Q}$ and $\mathbf{S}$ 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. Since the Q-matrix is a single skill per item then the type of Q-matrix does not affect the inference of the

result matrix.

$$\underset{\text{Items}}{\overset{\text{Examinee}}{\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}}} = \underset{\text{Items}}{\overset{\text{Skills}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}} \times \underset{\text{Skills}}{\overset{\text{Examinees}}{\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}}$$

$$R \qquad\qquad Q \qquad\qquad S$$

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, there remain many different solutions for this factorization.

Considering the large space of solutions to $\mathbf{R} \approx \mathbf{QS}$, it implies different algorithms may lead to different solutions. Many algorithms for matrix factorization search the space of solutions to equation (2.2) 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 $\mathbf{Q}$ or $\mathbf{S}$ (for NMF) is found by multiplying the current value by some factor that depends on the quality of the approximation in Eq. (2.2). It was proved that repeated iteration of the update rules is guaranteed to converge to a locally optimal factorization (**?**). 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 (2.3) 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 $\mathbf{R}$.

$$\|\mathbf{R} - \mathbf{QS}\|^2 = \sum_{ij} (\mathbf{R}_{ij} - \mathbf{Q}_i \mathbf{S}_j)^2 \tag{2.3}$$

Another cost function is based on the Kullback-Leibler divergence, which measures the divergence between **R** and **QS** as shown in equation (2.4).

$$D(\mathbf{R}||\mathbf{QS}) = \sum_{ij}(\mathbf{R}_{ij} log \frac{\mathbf{R}_{ij}}{\mathbf{Q}_i\mathbf{S}_j} - \mathbf{R}_{ij} + \mathbf{Q}_i\mathbf{S}_j) \tag{2.4}$$

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 **Q** and **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 (2.5). These update rules in equation (2.5) guarantee that the Euclidean distance $\|\mathbf{R} - \mathbf{QS}\|$ is non increasing during the iteration of the algorithm.

$$\hat{\mathbf{S}} \leftarrow \hat{\mathbf{S}} \frac{(\hat{\mathbf{Q}}^T\mathbf{R})}{(\hat{\mathbf{Q}}^T\hat{\mathbf{Q}}\hat{\mathbf{S}})} \qquad\qquad \hat{\mathbf{Q}} \leftarrow \hat{\mathbf{Q}} \frac{(\mathbf{R}\hat{\mathbf{S}}^T)}{(\hat{\mathbf{Q}}\hat{\mathbf{S}}\hat{\mathbf{S}}^T)} \tag{2.5}$$

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.

Barnes (**?**) proposed equation 2.6 for conjunctive model of Q-matrix 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 2.6, we will see that the $\neg\mathbf{R}$ matrix is a product of two matrices, **Q** and $\neg\mathbf{S}$

$$\mathbf{R} = \neg(\mathbf{Q}(\neg\mathbf{S})) \tag{2.6}$$

In next sections of this chapter the application of NMF on conjunctive type of Q-matrix and how this technique can derive a Q-matrix from a test result is given in details.

Besides its use for student skills assessment and for deriving a Q-matrix, matrix factorization is also a widely used technique in recommender systems. See for eg. **?** for a brief description of some of the adaptation of these techniques in recommender systems.

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

The other skills assessment models we consider are based on what is referred to as Deterministic Input Noisy And/Or (DINO/DINA) **?**. They also rely on a Q-matrix and they can not in themselves infer a Q-matrix from a test result matrix, and instead require a predefined Q-matrix for the predictive analysis. 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:

$$P(X_{ij} = 1 \mid \xi_{ij}) = (1 - s_j)^{\xi_{ij}} g_j^{1-\xi_{ij}} \tag{2.7}$$

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 (**?**).

2.5.4 Alternate Least-square Factorization (ALS)

The Alternate Least-Square Factorization (ALS) method is defined in **?** which is also a factorization method that can infer an initial Q-matrix to estimate its parameters, or refine a predefined Q-matrix given as a starting point. The next sections in this chapter introduce a methodology to refine a existing Q-matrix in more details.

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:

$$\hat{\mathbf{S}}_0 = (\mathbf{Q}_0^{\mathrm{T}} \mathbf{Q}_0)^{-1} \mathbf{Q}_0^{\mathrm{T}} \mathbf{R} \tag{2.8}$$

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:

$$\hat{\mathbf{Q}}_1 = \mathbf{R}\hat{\mathbf{S}}_0^{\mathrm{T}}(\hat{\mathbf{S}}_0\hat{\mathbf{S}}_0^{\mathrm{T}})^{-1} \tag{2.9}$$

And so on for estimating $\hat{\mathbf{S}}_1$, $\hat{\mathbf{Q}}_2$, ... Alternating between equations (2.8) and (2.9) yields progressive refinements of the matrices $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{S}}_i$ that more closely approximate $\mathbf{R}$ in equation (**??**). 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.

2.6 Recent improvements

Let us get back to the main contribution of the thesis: Assessing model fit on the basis of the goodness of fit measure with predictive performance analysis of real and synthetic data. The previous sections of this chapter introduced skills assessment techniques to obtain the predictive performance of a dataset. Some methods infer a Q-matrix and some other not even infer it but require a predefined one. Challenges are not limited to inferring a Q-matrix out of data: the number of latent skills is also unknown. For some real dataset there exists at least one pre-defined Q-matrix by an expert which could have potential mistakes and requires to be refined. These are some challenges that should be addressed as a complementary steps to predictive performance analysis.

The rest of this chapter presents researches that are complementary steps of the main contribution of this thesis. The general perspective of section 2.6.1 is to find a way for deriving the Q-matrix

from data, along with a Skills mastery matrix. Section 2.6.1 is inspired by **?** that was published in ITS conference. This article aims to find a method to derive these matrices for different types of Q-matrices. This is a valuable and challenging task. Finding the number of latent skills (i.e. the common dimension between matrices $\mathbf{Q}$ and $\mathbf{S}$) is another important task that is described in section 2.6.1 . The text of section 2.6.2 is mainly borrowed form **?** work that was published on EDM conference. Finally, in section 2.6.3 few methods is introduced to validate tasks to skills mapping which is also applicable for the refinement of a Q-matrix. Parts of section 2.6.3 is taken from **?** publication in EDM conference.

2.6.1 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 (**??**). **?** showed that different data mining techniques can extract item topics, one of which is matrix factorization. He showed that NMF works very well for synthetic data, but the technique's performance with real data was degraded. The above studis show 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.

? proposed an approach to successfully deriving a conjunctive Q-matrix from simulated data with NMF. The methodology of this research relies on simulated data. They created a synthetic data with respect to conjunctive model of Q-matrix. They proposed a methodology to assess the NMF performance to infer a Q-matrix from the simulated test data. This methodology is conducted by comparing the predefined Q-matrix, $\mathbf{Q}$ which was used to generate simulated data with the $\hat{\mathbf{Q}}$ matrix obtained in the NMF of equation 2.6.

As expected, the accuracy of recovered Q-matrix degrades with the amount of *slips* and *guesses* which are somehow noise factor in their study. They showed that 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.

The results of their experiment have few advantages that are lost with real data. The most important ones are : the number of skills is known in advance and also the Q-matrix is predefined which makes it easier to validate the results. On real data NMF could not recover the expert defined Q-matrix properly. This can have few reasons that bolds our main contribution. The first reason is that the model that we are testing to infer a Q-matrix is not a fit for the dataset to describe it into its

parameters. The second reason could be the variety of defining Q-matrix by different experts and also possible mistakes that experts can have in defining a Q-matrix.

2.6.2 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 (**?**) investigated 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 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 results of this experiment 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. This also could show the importance of model selection. There is a possibility that the ground truth of the real dataset we used in this experiment is not the linear conjunctive model.

2.6.3 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 (**?????**). **?** compared 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.Two techniques for this purpose rely on the DINA and DINO models, whereas one relies on a matrix factorization technique called ALS.

To validate and compare the effectiveness of each technique for refining a given Q-matrix, they

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. The results of this experiment show that all techniques could recover alterations but the ALS matrix factorization technique shows a greater ability to recover alterations than the other two techniques.

2.6.4 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 2.3. Adding an other value to this cost function can lead to a new equation 2.10.

$$\|\mathbf{R} - \mathbf{QS}\|^2 + \kappa(\mathbf{O}\hat{\mathbf{R}}) \tag{2.10}$$

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 outcome of this experiment is the model specific parameters of the two synthetic generated datasets by each of the three tech-

niques. The final result of this experiment showed that when the synthetic data is generated based on POKS model, the Q-matrix could not perfectly inferred by the NMF and combined techniques. The reverse is also true, When the data is generated based on a conjunctive type of Q-matrix, then the estimated knowledge structure parameter of POKS model have no simple interpretation.

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 than a linear generated dataset with a conjunctive type of Q-matrix. 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 highly depends on the underlying model of the dataset. This is where the second contribution was created 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. 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

? 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 **?** 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 playing an increasingly important role in EDM (**?**). The availability of new simulation software is contributing to this trend in every domain. As an example **?** used simulated data for disease outbreak detection where simulated data is generated from a hypothesized model of a phenomena. The framework of this study relies on synthetic data. Getting back to the main contribution of this thesis, we proposed an approach for model selection that relies on 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. Then the process of generating data should be accurate such that both data and model specific parameters are applied properly. Every model studied here can be considered *generative*, to the extent that they can generate synthetic data.

A unique advantage of synthetic data is that the model parameters can be predefined. Once the simulated data is generated with predefined parameters, a model can be trained over the generated data and a comparison with the original, known parameter values becomes possible. This may be the strongest benefit of synthetic data in assessing model fit. Since the hypothesis of this research relies on comparison the performance vector of real vs. synthetic data, then we can also consider data specific parameters of the real data in the generation process to make a better comparison of the results.

4.1 Data generation parameters

As described before, there exists two types of parameters for student test outcome:

— Model specific: These parameters are dependent on the ground truth of a dataset. Skills assessment models that were described in section 2.2 considers different parameters. Depending on the type of student model, some of them may share a parameter such as Q-matrix for multiple skills models but still the interpretation of each model is unique. Generally an experiment should be designed to learn these parameters form a dataset or they can be generated under specific criteria. More details for each parameter is given later in this chapter.

— Data specific (contextual): Regardless of the ground truth of a dataset, there are parameters that describe the dataste's specifications such as number of students. These parameters are common for any student model. They could possibly affect the performance of a model over a dataset.

Table 4.1 shows a complete list of both types of parameters required for data generation.

Contributed skills		Skills Model	Parameters: Model specific		Parameters: Data specific
Multiple		NMF Conj.		• Q-matrix	• Number of students • Number of items
		NMF Add.			
		DINA	• Slip • Guess	• Students skills mastery matrix	
		DINO			
Single		IRT	• Student ability • Item difficulty • Item discrimination		• Number of skills • Test success rate
		Expected	• Student Odds • Item Odds		
Zero		POKS	• Initial Odds • Odds ratio • PO structure		• Student score Variance • Item score Variance

Table 4.1 Parameters involved in synthetic data generation

4.1.1 Assessing parameters

To generate artificial data a set of parameters is required There are two approaches to assess required parameters to generate synthetic data:

— Parameters borrowed form a real data: Deriving contextual parameters from a real data is a straight forward task. To get model specific parameters we need to learn them from a real dataset given a model. In some cases there exists a predefined (or expert defined) parameter associated with the real data such as expert defined Q-matrices.

— Generate parameters randomly based on a distribution: In the case that there exists no reference dataset to derive parameters we can randomly select some samples based on a distribution (for example normal distribution). Some parameters have specific conditions that if they become violated it changes the definition of the parameter. More details is given in next sections.

The rest of this chapter describes the data generation process based on each model assessment technique that is used in the proposed model selection approach.

4.2 POKS

POKS is a technique that categorized as a “zero skills” method does not directly attempt to model underlying skills but instead rely on observable items only.

4.2.1 Obtaining parameters

This model has three parameters with which the inference in possible : Knowledge structure, Initial probabilities, Odds ratio. To extract these parameters from an existing data we use POKS [c0] R package to learn all the parameters given a dataset. In the case that the data in not available we will generate the parameters first. Below we will describe how these parameters are generated with different conditions and constraints.

Knowledge structure(KS)

This parameter shows the relation and dependency among a set of items. In fact, it reflects the order of learning a set of items in the same problem domain. Assuming items as different "nodes" in the structure there is a kind of "parent-child" relation between pairs of nodes where an easier item can be the child of a harder item. In our study this relation and dependency is considered as a "link" between pair of items. This link is not necessarily exists between each pair. Only those items that are related in the learning process can have a pairwise link. The graphical demonstration of knowledge structure is an directed graph with no cycles allowed, except for symmetric relations. Also in the current version of POKS transitive relations between items are explicitly ignored.

There are few other parameters involved in the generation of KS which are:

— Number of independent graphs: Even in a set of items there could be different groups of nodes without any dependency between them. Since there is no link between each group them they create an in dependent graph in the knowledge structure. This parameter can also affect the total number of links in a structure where more independent groups results in less number of links in the structure given same density of links in each group.

— Total Number of links: This has a correlation with the previous parameter but still we can control the number of dependencies in a KS. This parameter has an direct effect on test success rate and item score variance. Fewer number of links will result in lower test success rate and item score variance.

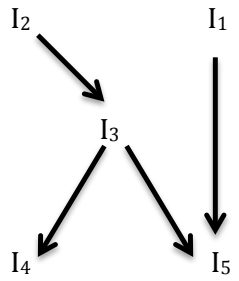
Adjacency matrix is one way to represent the KS. To avoid having a cycle in the structure we randomly assign 0-1 values to an upper triangular of this matrix. For simplicity we consider the structure to be a single connected graph and the number of links corresponds to half of cells of the upper triangular of this matrix. figure **??** shows a adjacency matrix associated with its graphical structure.

Once the KS is generated we should assign proper value to each item as an initial value and to each pair of items as odds ratio. One of the important concepts in assigning values to initial probabilities

c0. http://www.professeurs.polymtl.ca/michel.desmarais/Papers/UMAP2011/lib-poks.R

$$\begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{array} \begin{array}{c} \begin{array}{ccccc} i_1 & i_2 & i_3 & i_4 & i_5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

(a) Adjacency matrix



(b) Graphical representation

Figure 4.1 An example of random KS with 5 items

and odds ratio (explanation is in the next subsection) is the "level" of a node in the graphical representation of the KS which corresponds to the longest path from that node to a leaf node. A leaf node is a node that does not have any child and a root is a node without any parents. Equation 4.1 shows the definition of this function where $length-of$ returns the length of a path between two nodes and $path$ returns true value if there exists a directed path between two nodes and $leaf$ returns true if the node is a leaf. For example in figure **??** I_5 is a leaf with the level of 1 and I_2 is a root with level of 3.

$$L(KS,N_i) = max(E|E = length-of(N_i,p), \exists path(N_i,p), leaf(p) = True) \tag{4.1}$$

Initial Probabilities

Each item should be associate with an initial probability of success. This probability has a direct correlation with the difficulty of the item. In general if node A implies node B ($A \rightarrow B$) then node B should have a higher initial probability of success because in this relation item A is the harder problem. The allocation of initial values to items should be done respect to this characteristic.

Starting from a root node in KS we assign a random initial probability for each node N_i from a range of values between r_{max} to $r_{max} + \frac{1-r_{max}}{L(KS,N_i)}$ where r_{max} is the maximum initial probability of N_i parents. For a node that is a root r_{max} is 0 since it is starting node and obviously should have the lowest initial probability among its descendant. Also for a leaf the assignment range will be between r_{max} and 1 because it has level of 1.

As mentioned before, the initial probability of success for items come from these values. This set of values changes when a sample is assigned to each item. The process of sampling preforms item by item and in each step a new set of values are assigned to items as the probability of success. This set of probabilities is called the "state" of probabilities (shown as S where S_i is the probability of

success for item *i*). In this context we use odds instead of probabilities where $O(H) = \frac{P(H)}{1-P(H)}$. The details for changing the "state of odds" and assigning samples to items is given in next sections.

Odds ratio

The very last parameter is the odds ratio which is a ratio that represents the strength of a link between a pair of items. In the inference for POKS model this parameter is used to update the initial probability of inferred items given a set of observations. The same situation as inference we use this parameter to update the state of odds once an item has been sampled (more on these implementation details is given in the next section). For inference in POKS, the probability update for node H given E_1,... E_n can be written in following posterior odds form:

$$O(H|E_1, E_2, ..., E_n) = O(H) \prod_i^n \frac{P(E_i|H)}{P(E_i|\overline{H})} \tag{4.2}$$

where odds definition is $O(H|E) = \frac{P(H|E)}{1-P(H|E)}$ and $O(H)$ is the initial odds of node H. 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. we will follow the same steps to generate data based on POKS model. There are two types of odds ratio in the POKS model. Assuming I_3 as a node that has two children and a parent in figure **??**. if the sampling result for this item (with respect to its initial odds) becomes 1 (or success) then we should update the odds of its children because a success for a harder item should increase the chances of success for an easier one. Therefore we define a "true odds ratio" which is applicable when the evidence (in this case the evidence is the one that has been sampled, node I_3) becomes a success. If the evidence is 0 (or failure) then we should update the odds of its parents since a failure for an easier problem should decrease the chances to success a hard one. To update the parents' odds we use "False odds ratio" when the evidence is false. In fact $\frac{P(E_i|H)}{P(E_i|\overline{H})}$ in equation 4.2 represents the odds ration for item H given evidence E_i.

In our experiments we assigned random values greater than 0.5 as "true odds ratio" and smaller than 0.5 for "false odds ratio"for those pairs of items that have link in KS.

4.2.2 Data generation

Previous sections explained the process of obtaining a pre-defined or a random generated set of parameters for POKS. In this section we explain the process for sampling data points values to create an artificial student test result matrix given a set of parameters. Each record in this approach represent a student test result that requires n iterations to be generated where n is the number of items and the final result matrix needs m (number of students) records. The process simply follows steps in algorithm 1. Line 8 to 15 of this algorithm updates the state probability of the items. It is

important to note that we just look at the neighbors of the item that is sampled. In other words the update propagate in breadth.

Algorithm 1 POKS data generation

for each record $i \in m$ (number of students) **do**
    or.f = False Odds Ratio
    or.t = True Odds Ratio
    S = Initial odds
    for each item $j \in n$ (number of items) **do**
        Randomly pick item j that has not been sampled
        R_{ij} = Sample item j for record i with respect to S_j
        if $R_{ij} = 1$ **then**
            $U =$ children of item j in KS
            $\forall c \in U : S_c = S_c \times or.t_{cj}$
        end if
        if $R_{ij} = 0$ **then**
            $U =$ parents of item j in KS
            $\forall p \in U : S_p = S_p \times or.f_{pj}$
        end if
    end for
end for

4.2.3 Data specific parameters

Obtaining the performance of different models over a dataset is a main part of our contribution in this study. Some models require number of skills to learn their parameters such as DINA. Since POKS generated dataset does not directly attempt to model underlying skills we have to have a prediction for this important parameter to associate with the data. **?** proposed two approaches based on factorization technique to predict an optimum number of skills given a dataset.

The only way to control the test result success rate and student/item score variance is to apply changes on the initial odds of items which are used to sample student test result. For student scores variance, for each record 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 but still the distribution of items score variance remains the same. Changing the initial odds that follows a specific distribution will create a dataset in which the item variance is following that distribution. It is important to note that this change should not violate any relation in the KS. 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.

4.3 IRT

Equation 4.3 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.

$$P(X_j = 1 \mid \theta) = \frac{1}{1 + e^{-a_j(\theta - b_j)}} \tag{4.3}$$

$$-4 < \theta < 4$$

$$-3 < b_j < 3$$

[what about a_j?]

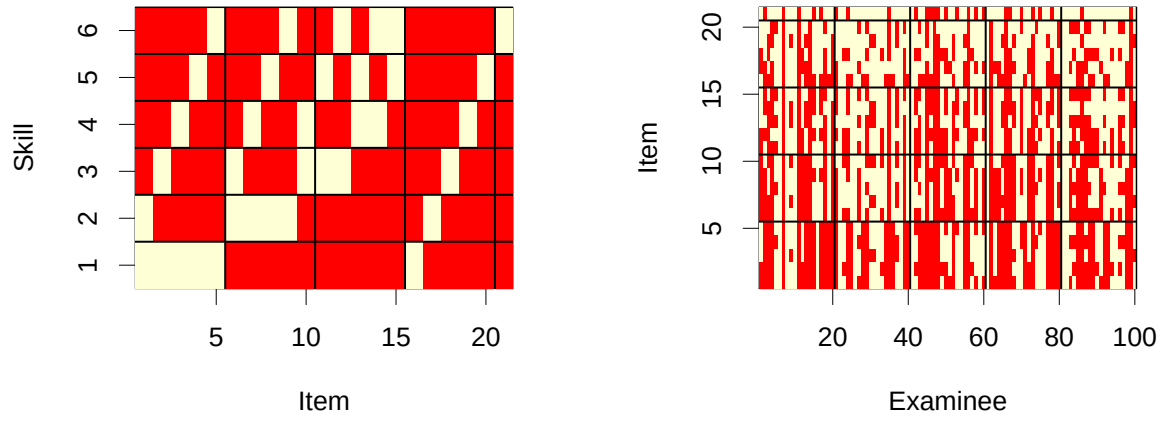
[Actually, it would be better to write this with equations, including the Poisson distribution. See this paper (equation 2 and below): http://educationaldatamining.org/EDM2011/wp-content/uploads/proc/edm2011_paper35_full_Desmarais.pdf]

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 [why these numbers? Shouldn't they be from the sample]. 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.

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.

4.4 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 4.2. 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.



(a) Q-matrix of 6 skills

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

Figure 4.2 Q-matrix and an example of simulated data with this matrix. pale cells represent 1's and red ones represent 0's.

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 skills mastery.

Once the Q-matrix and Skills mastery matrix is created we can produce the test result matrix with equation (2.6). 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 4.2 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.

4.5 Linear NMF Additive

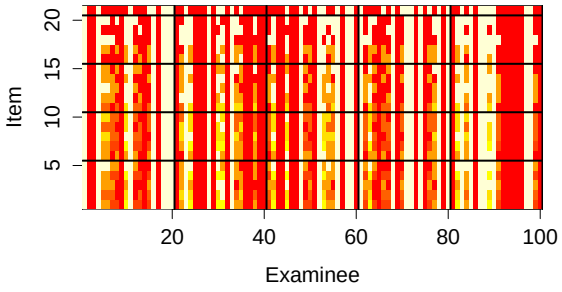
The process to create synthetic data based on additive type of Q-matrix is almost the same as Conjunctive one. The difference is on the interpretation of the Q-matrix that changes the step where the result matrix is producing.

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.

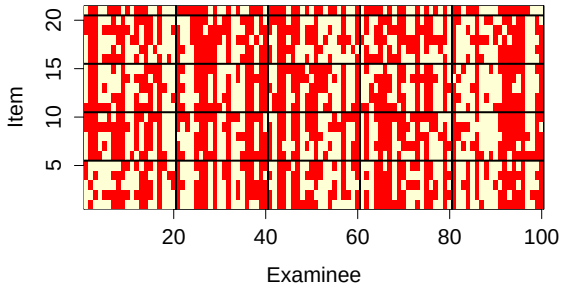
items

skills																	
	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

(a) Additive Q-matrix



(b) Raw Result matrix



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

Figure 4.3 Additive model of Q-matrix and Corresponding synthetic data

Figure 4.3(a) shows an additive type of Q-matrix and figure 4.3(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.

4.6 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 4.4 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.

$$P(X_{ij}=1 \mid \xi_{ij}) = (1-s_j)^{\xi_{ij}} g_j^{1-\xi_{ij}} \tag{4.4}$$

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.

4.7 Educational data generator

With all the parameters and models that were described, it is still challenging to generate synthetic data for a specific model with a set of given parameters. We (**?**) create a package that generates artificial data under a specific model's assumption given a set of parameters. Previous sections described the generation process for each model. This task is easy and possible as long as two conditions are satisfied :

— First: All the parameters for a model are given. Table 4.1 shows a list of required parameters to generate artificial data.

— Second: There exists no conflict between the given parameters. For example given "number of skills" as a parameter equal to 4 and a predefined "Q-matrix" with 5 skills is a conflict to generate a synthetic data based on DINO model. However, testing these conflicts is not an easy task and also resolving a conflict is another challenge.

There exists a hierarchical structure among parameters of a model. Some lower level parameters can generate more complex ones which are included in the set of parameters that directly generates data. This process requires default values and distributions. Namely, given the number of items, skills and students, the Q-matrix and skills mastery matrix can be generated with a default normal distribution on items difficulty and students ability. Hence, generating a dataset based on NMF conjunctive model will be possible since the prerequisite parameters are available. Therefore there are different combination of the sufficient parameters to generate a data based on a model. For example for a synthetic data with DINO model this set can be Q-matrix, Skills mastery matrix, Slip and Guess vectors or Number of skills, Items and students.

Given the tow conditions are satisfied, there would be different ways to generate data depending on the available set of parameters

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 the 6 models reviewed in chapter 2 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 us 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, table 5.1 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 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 2.2 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.

5.3 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 5.1. 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.

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.

5.3.1 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

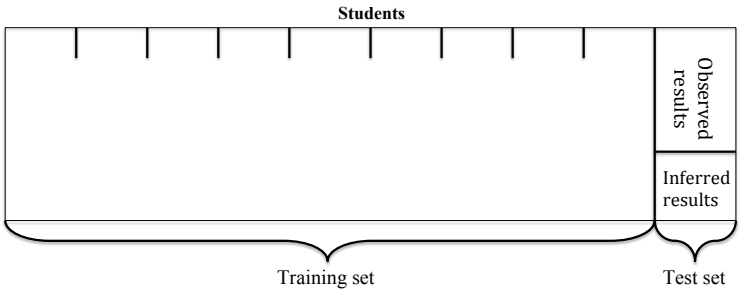


Figure 5.1 Data breakdown of cross validation process

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	
Test success rate	0.25 to 0.85	
Student score variance	0.03 to 0.20	
Item score variance	0.03 to 0.20	
Item discrimination	0.5 to 3	IRT
Item difficulty	-3 to 3	
Student ability	-4 to 4	
Simulation parameters		
Number of observed items	9 to Number of items -1	All models
Training set size	90% of $[100-1500]$	
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		Observed items
Multiple	NMF Conj.		• Q-matrix	• Students skills mastery matrix
	NMF Add.			
	DINA	• Slip • Guess		
	DINO			
Single	IRT	• Item difficulty • Item discrimination		• Student Ability
	Expected	• Item Odds		• Student Odds
Zero	POKS	• Initial Odds • Odds ratio • PO structure		

Table 5.3 Parameters of the predictive performance framework

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.

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 (2.2) were generated and the same process was used for both the DINO and DINA data sets. Item outcome is then generated according to equation (2.7) 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 (2.2) 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-

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

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 2.5.4, starting with an initial random Q-matrix.

5.3.2 Predictive performance results

This section shows the result of predictive performance of the seven models over the 14 datasets described in table 5.4.

Figures 5.2 and 5.3 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.

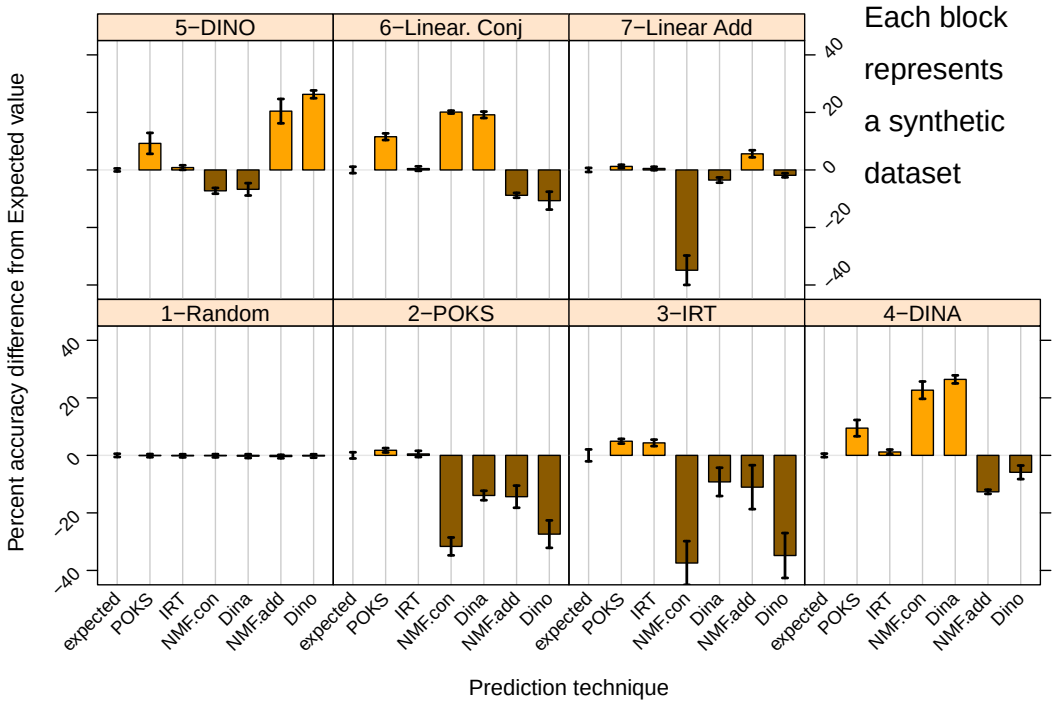


Figure 5.2 Item outcome prediction accuracy results of **synthetic data sets**

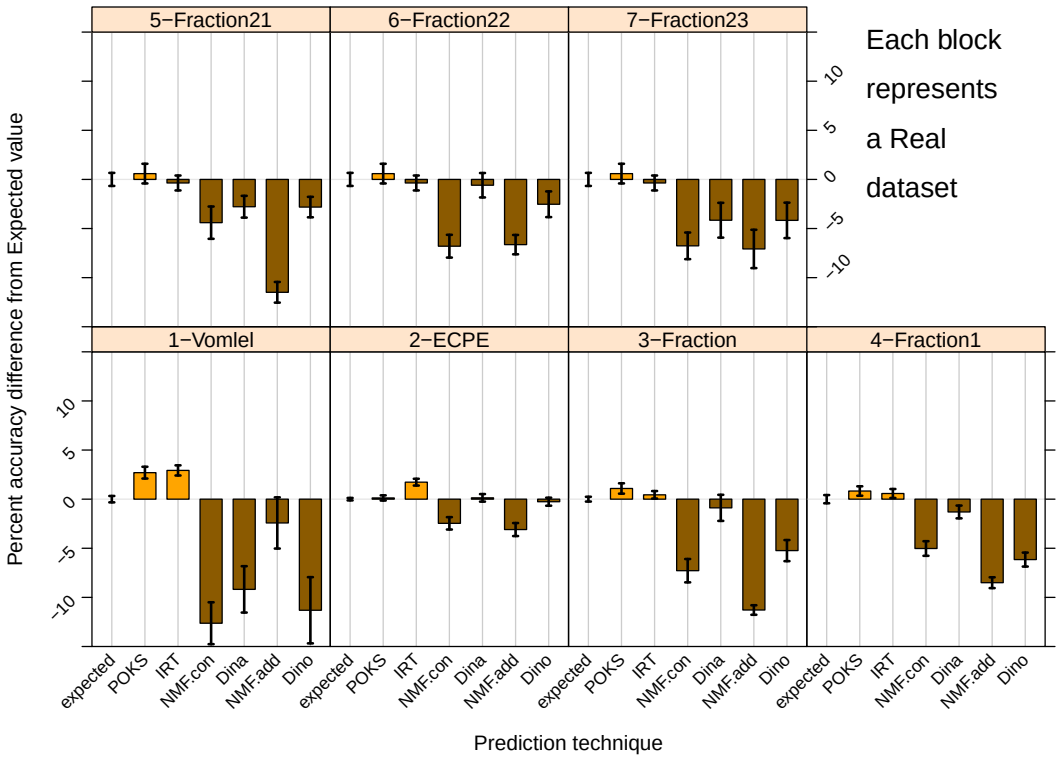


Figure 5.3 Item outcome prediction accuracy results of **real data sets**

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.

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 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.

5.4 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 5.4 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 5.5 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 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 5.6 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 5.7 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 performance values shifts across correct prediction axis but the pattern stays the same.

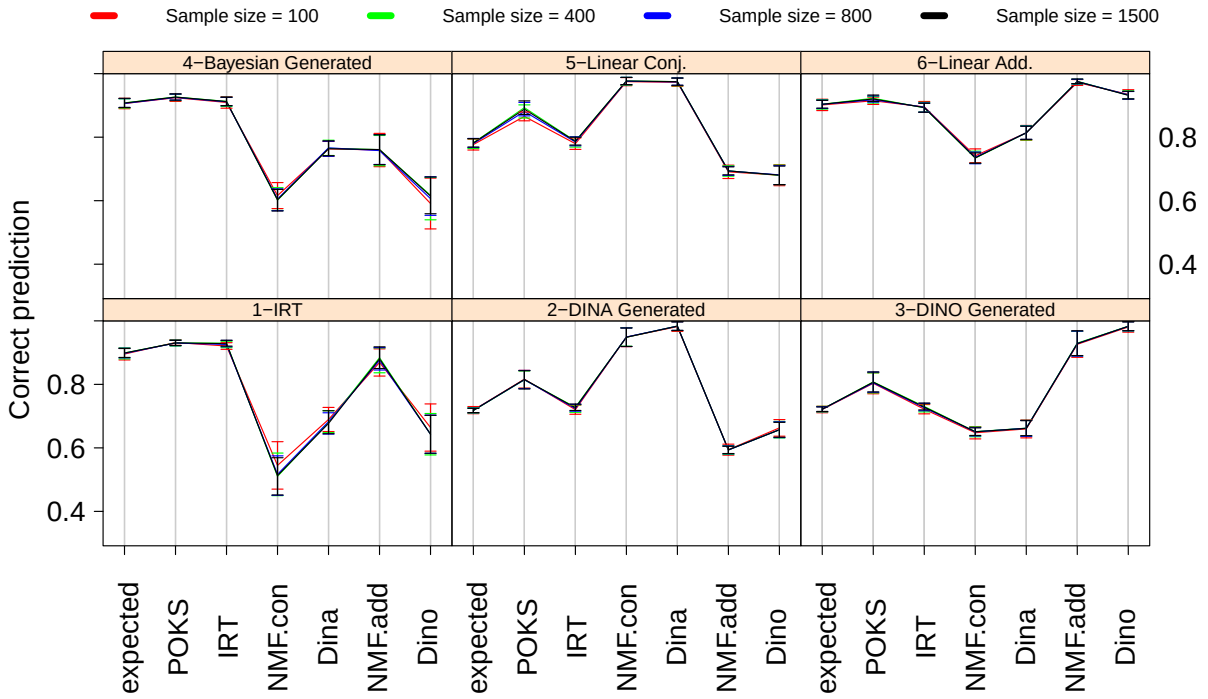


Figure 5.4 Variation of **Sample Size** Over synthetic data sets

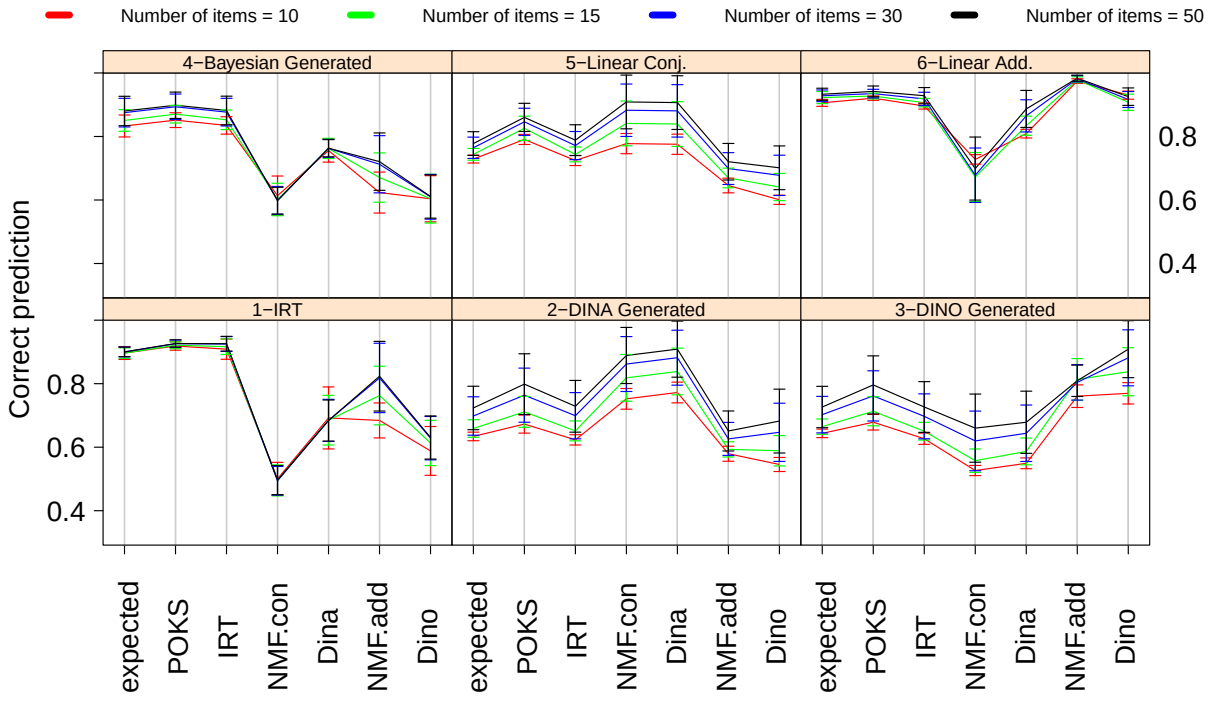


Figure 5.5 Variation of **Number of items** Over synthetic data sets

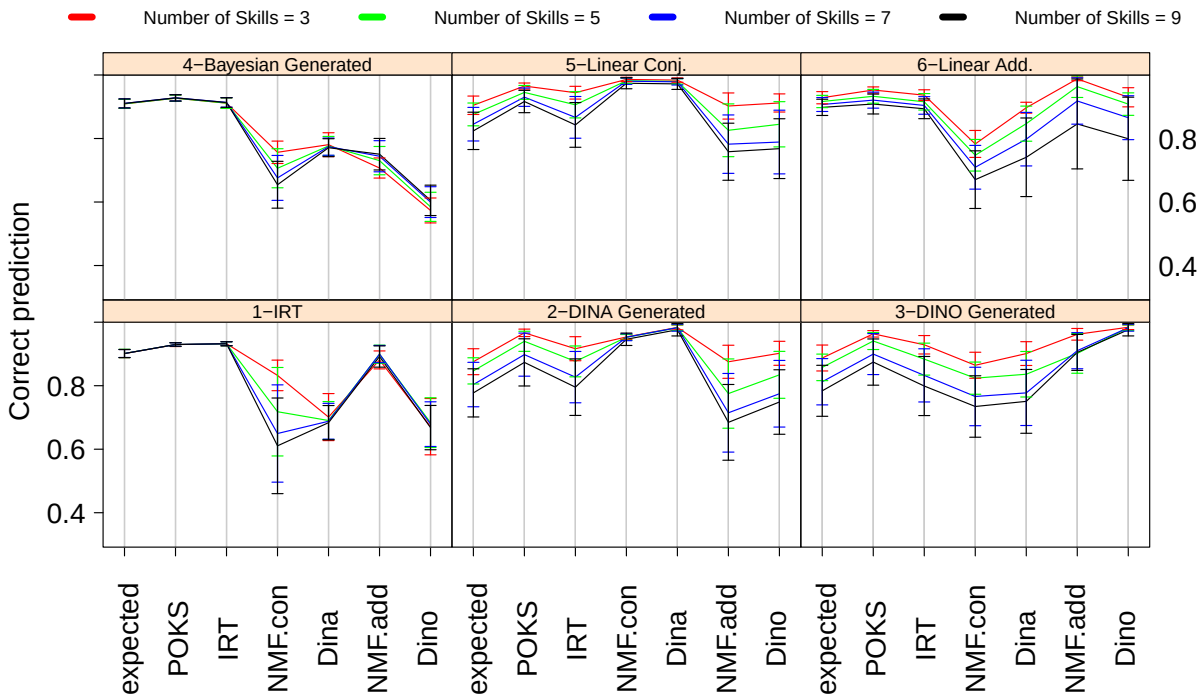


Figure 5.6 Variation of **Number of skills** Over synthetic data sets

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.

5.4.1 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

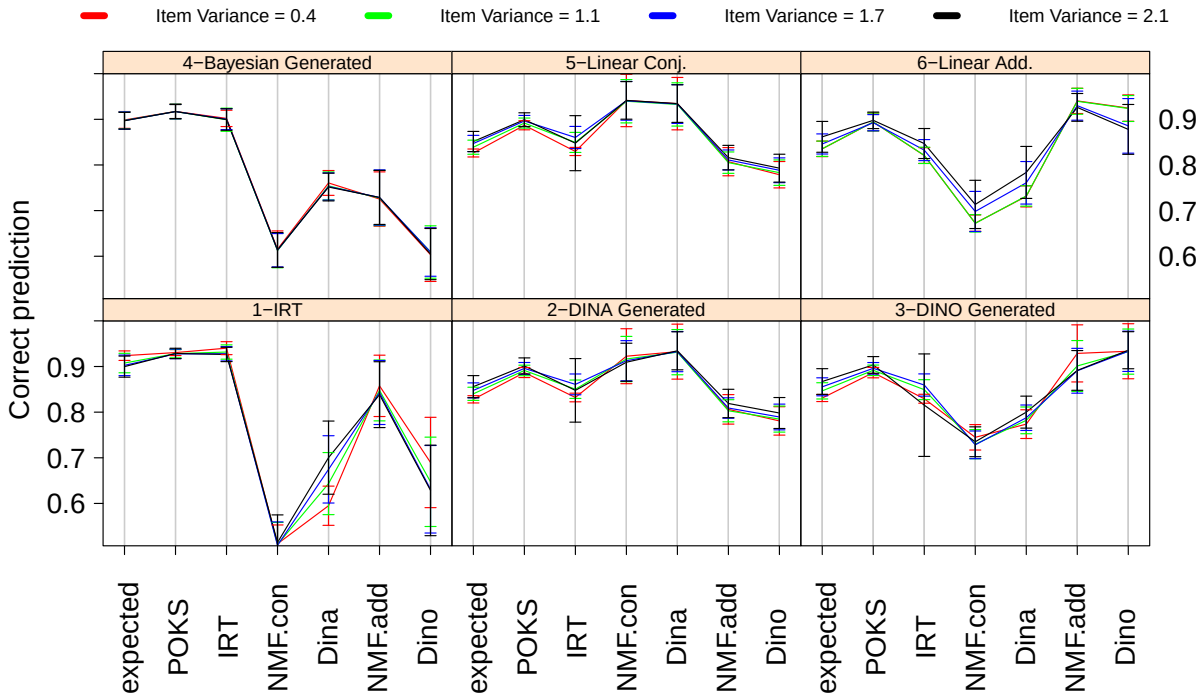


Figure 5.7 Variation of **Item Variance** Over synthetic data sets

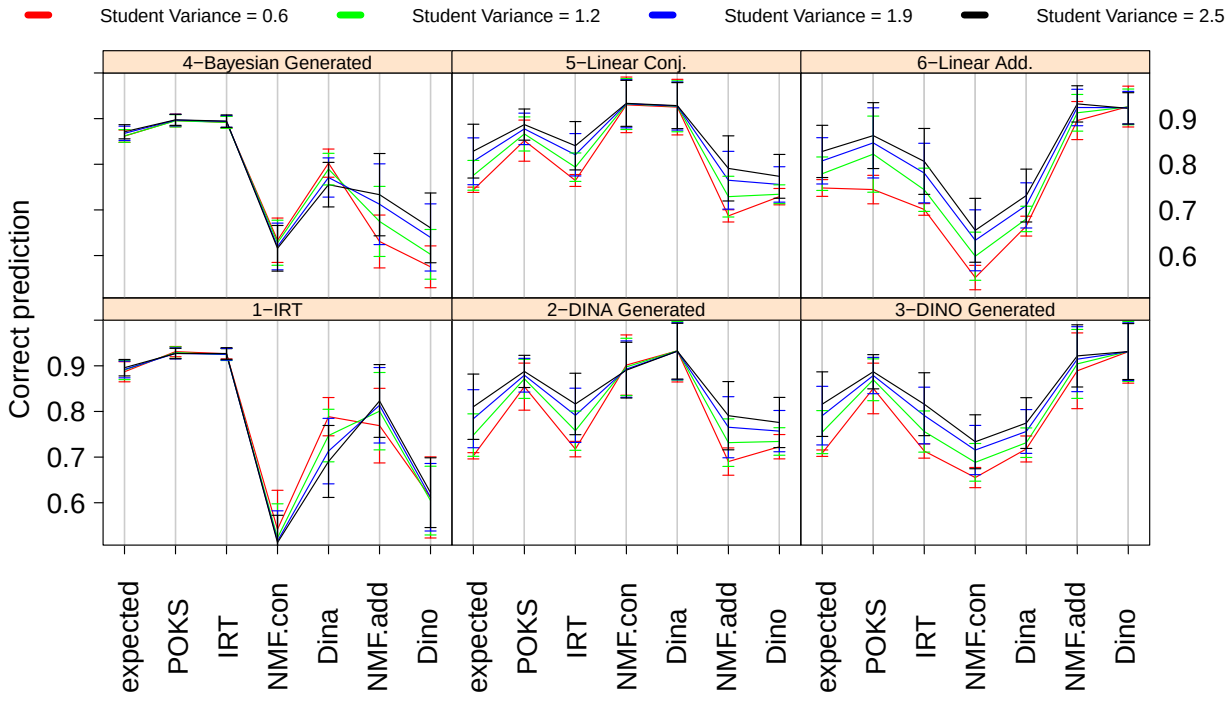


Figure 5.8 Variation of **Student variance** Over synthetic data sets

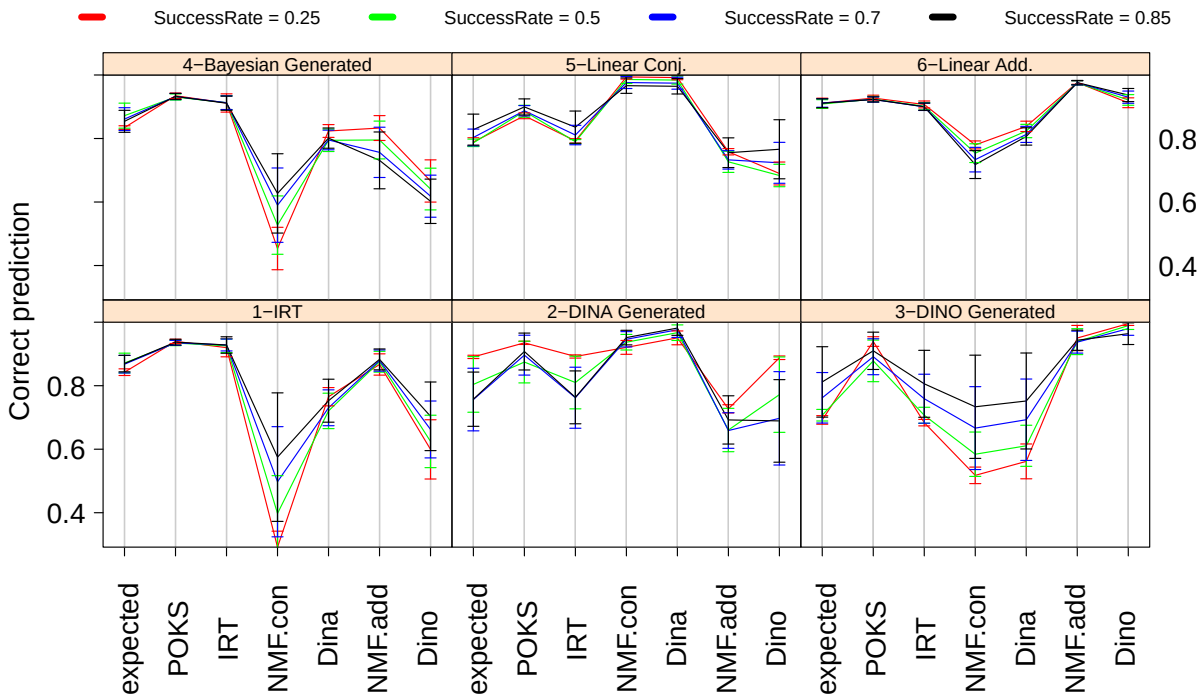


Figure 5.9 Variation of **Success Rate** Over synthetic data sets

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 5.4) 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 disjunctive model then it's less correlated.

		Synthetic Datasets					
		POKS	IRT	NMF Conj.	DINA	NMF Add.	DINO
Synthetic Datasets	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			
		Vomlel	ECPE	Fraction	1	21	22	23
Synthetic Datasets	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

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 5.4 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 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 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

BP: Best Performer
NN: Nearest Neighbor

		Datasets												Accuracy (%)	
		POKS		IRT		NMF Conj.		DINA		NMF Add		DINO			
		BP	NN	BP	NN	BP	NN	BP	NN	BP	NN	BP	NN	BP	NN
Models	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

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 if 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