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. GUÉHÉNEUC Yann-Gaël, 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 addressed in this thesis: how to decide which are the most representative of the underlying ground truth.

The standard practice is to train different models, and consider the one with the highest predictive performance as the best fit. However, model performance typically varies along factors such as sample size, target variable and predictor entropy, noise, missing values, etc. For example, a model's resilience to noise and ability to deal with small sample size may yield better performance than the ground truth model for a given data set.

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 of model performances, and use a nearest neighbor approach with a correlation distance to identify the ground truth model. This approach is based on the following definitions and procedure. Consider a set of models, $\mathscr{M}$, and a vector $\mathbf{p}$ of length $|\mathscr{M}|$ that contains the performance of each model over a given data set. This vector represents a point that characterize the data set in the performance space. For each model $M \in \mathscr{M}$, we determine a new point in the performance space that corresponds to synthetic data generated with model M. Then, for a given data set, we find the nearest synthetic data set point, using correlation as a distance, and consider the model behind it to be the ground truth.

The results show that, for synthetic data sets, their underlying model sets are generally more often correctly identified with the proposed approach than by using the best performer approach. They also show that semantically similar models are also closer together in the performance space than the models that are based on highly different concepts.

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
RMSE	Root Mean Square Error
SSE	Sum Square Error

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 (Domingos and Pazzani, 1997).

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:

— **Student model**: "In general terms, student modeling involves the construction of a qualitative representation that accounts for student behavior in terms of existing background

knowledge about a domain and about students learning the domain. Such a representation, called a student model" (Sison and Shimura, 1998). Student skills assessment models are essentially constructed to assess student's skills or estimate potential skills required for problems. Since our experiments are in the domain of educational data mining, by the term "Model" we mean "student model".

— **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 has 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 which are also known as "Contextual parameters" 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. Note that for the skills models studied here, the ground thruth is always unknown, unless we use synthetic data.

— **Performance of a model**: The accuracy of a model to predict student response outcomes over a dataset (using cross-validation) is called performance of a 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 has a performance over a dataset. Consider a set of models, $\mathscr{M}$, and a vector $\mathbf{p}$ of length $|\mathscr{M}|$ that contains the performance of each model over a given data set. This vector represents a point in the performance space, and it is defined as the *performance vector* of that data set.

— **Performance Signature**: The *performance vector* can be considered from two perspec-

tives: 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 **performance signature** for a specific data which considers the vector in a single dimensional space which is essentially a line representing performances of the candidate models. They are sharing the same concepts but different presentations. We will refer to **performance signature** for the purpose of easier visualization in a single dimensional space.

— **Performance Prototype**: We use the term *performance prototype* to designate the *performance vector* associated with the synthetic data of a model class. Note that there are different ways to produce the synthetic data of a model and we return to this question later.

— **Target Performance Vector**: This term is used to refer to the *performance vector* of the real data set to classify.

1.3 Research Questions

The following questions are addressed in this thesis:

1. What is the *performance vector* of student skills assessment models over real and over synthetic data created using the same models?
2. Is the *performance vector* unique to each synthetic data type (data from the same ground truth model)?
3. Can the *performance vector* 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 *performance vector* of the synthetic and real data. The third objective is to test the uniqueness and sensitivity of the *performance vectors* 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 *performance vectors* of two datasets with the same ground truth have a high level of correlation across different data specific parameters.

Hypothesis 2: The best performer model is not necessarily the ground truth model.

Hypothesis 3: Datasets with the same model parameters and data specific parameters create unique *performance vector*.

1.6 Main Contributions

The main contribution of this thesis is assessing model fit of a data set by comparing its *performance vector* to the *performance vectors* of synthetic datasets generated with the same data specific parameters of the given dataset but skills assessment models. 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 “performance signature” to contextual factors should be considered in the comparison of the *performance vectors*. The other contribution is to test the stability of the “performance 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 **??** discusses recent work about model selection. The main contribution of the research starts from chapter 3 where we explain the proposed approach in details. As a complementary part of the proposed approach we explain synthetic data generation approaches in chapter 4. The experimental results of the main contribution are 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 and they are a fundamental part of intelligent learning environments (Desmarais and Baker, 2011). 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 in Koedinger et al. (2011)), 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 test outcome prediction. Student models incorporate single or multiple skills. Some even model performance without explicit skills and we will refer to them as zero-skill models. The most widely used one is Item Response Theory (IRT). In its simplest version, IRT considers a single skill for 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 are directly observable among 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 ways in which each skill can contribute to succeed a problem: The first case is when mastering a skill is mandatory for a student to succeed an item that requires it. 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 $+4$. Some 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 later in this chapter.

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 (Tatsuoka, 1983, 2009). Figure 2.1 shows an example of this mapping which is named Q-matrix. Figure 2.1(b) 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(c).

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 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 (Feng et al., 2009) have developed an intelligent tutoring system (the ASSISTment system) that relies on fine grain skills mapped to items. Barnes, Bitzer, & Vouk in (Barnes et al., 2005) 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: $k < nm/(n+m)$ (Lee et al., 1999) where k , n and m are number of skills, students and items respectively.

s_1 : fraction multiplication

s_2 : fraction addition

s_3 : fraction reduction

(a) Skills

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

(b) Items with different skills

Skills

Items	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

(c) Q-matrix

Figure 2.1 Four items and their corresponding Q-matrix

The skills mastery matrix, **S**, 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 a 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 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.

— **Additive** (compensatory): Compensatory or additive model of skills is an interpretation of a Q-matrix where each skill contributes some weight to succeed that item. For example, if an item requires two skills a and b with the same weight for each of them, 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.

Note that in practice these types of interpretations becomes reasonable when there is at least one item in the Q-matrix that requires more than one skill otherwise they preform the same as each other. Later in this chapter these types will be described in more details with examples.

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.

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 Desmarais and d Baker (2012), 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.

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 from this category.

— 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

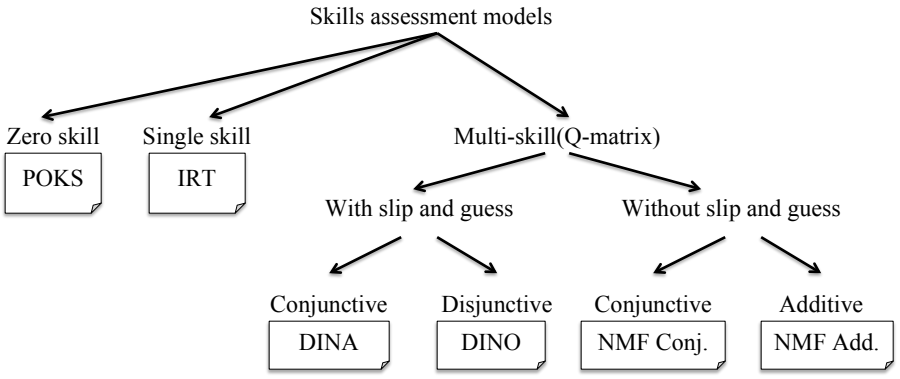


Figure 2.2 Skills assessment methods

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 also 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 (Doignon and Falmagne, 1999; Desmarais et al., 2006), 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 (Desmarais et al., 1996a). 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.

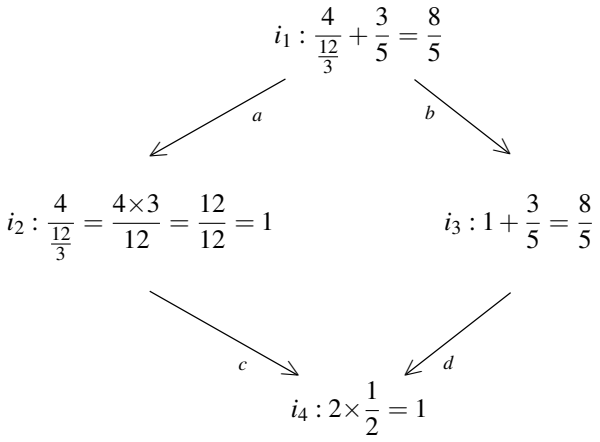


Figure 2.3 Partial Order Structure of 4 items

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 (Doignon and Falmagne, 1985). In POKS, this constraint is relaxed 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 (Desmarais et al., 1996b; Desmarais and Pu, 2005).

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.

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

Figure 2.4 Oriented incidence matrix and Adjacency matrix

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.

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 (Baker and Kim, 2004). The second approach is a trivial approach we call **Expected Prediction**. This approach is also 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) (Baker and Kim, 2004). Each item has a difficulty and a discrimination parameter.

IRT assumes the probability of success to an item X_j by student i is a function of a single ability factor θ_i:

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

In the two parameter form above, referred to as IRT-2pl, where parameters are:

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

Student ability, 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.

A simpler version of IRT called the Rash model, fixes the discrimination parameter, a, to 1. Fixing this parameter reduces over fitting, as the discrimination can sometimes take unrealistically large values. Rash model is a valid model for student skills modeling, but we do use the more general IRT-2pl model, which includes both a and b, for the synthetic data generation process in order to make this data more realistic (chapter 4 discusses data generation approaches in details).

2.4.2 Baseline Expected Prediction

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

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

where $O(X_i)$ is the odds of success to item i by all participants and $O(S_j)$ is the odds of success rate 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: $P(X) = (f(x=1)+1)/(f(x=1)+f(x=0)+2)$, where $f(x=\{1,0\})$ represents the frequency of the corresponding category $x=1$ or $x=0$.

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 this category of models can be divided into two sub-categories:

— 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 is 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. (Cen et al., 2005, 2006) 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 (Barnes, 2006) 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 for 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.

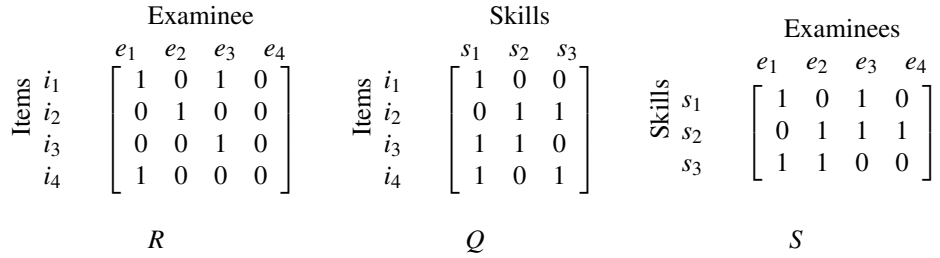


Figure 2.5 An example for Conjunctive model of Q-matrix

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 that 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 a Q-matrix, 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.

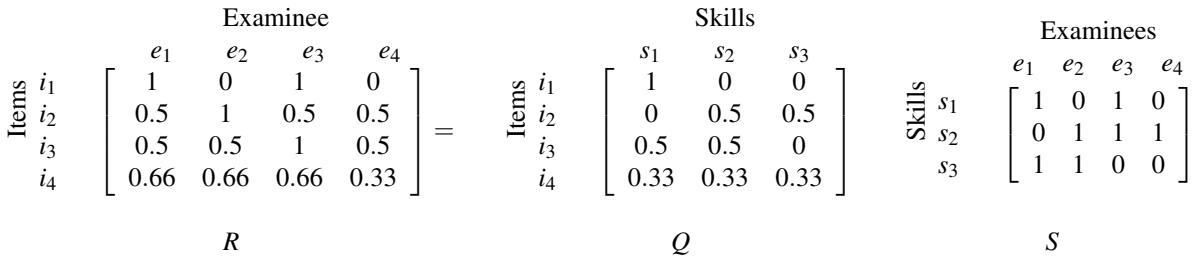


Figure 2.6 An example for Additive model of Q-matrix

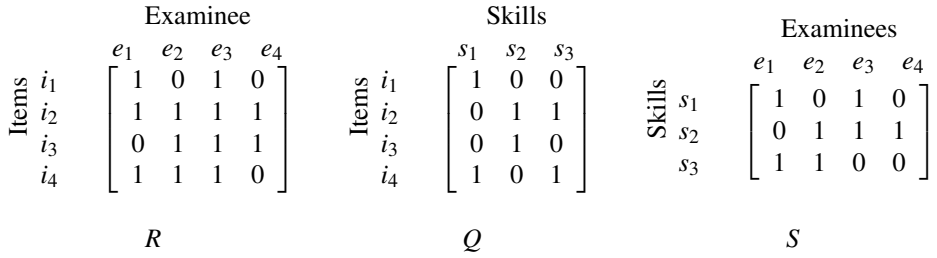


Figure 2.7 An example for Disjunctive model of Q-matrix

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. (Thai-Nghe et al., 2011) 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 (Corbett and Anderson, 1995). 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.

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

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

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 (Seung and Lee, 2001). We refer the reader to (Berry et al., 2007) 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 $\mathbf{R}$ and $\mathbf{QS}$ 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 $\mathbf{R} = \mathbf{QS}$ (Seung and Lee, 2001). For simplicity we just consider the cost function based on the Euclidean distance.

The gradient descent algorithm used to minimize the error is iterative and in each iteration we expect a new estimation of the factorization. We will refer to the estimated Q-matrix as $\hat{\mathbf{Q}}$ and the estimated Skill mastery matrix as $\hat{\mathbf{S}}$. The iterative gradient descent algorithm should change $\mathbf{Q}$ and $\mathbf{S}$ to minimize the cost function. This change should be done by an update rule. Lee and Seung (Seung and Lee, 2001) 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 (2005) 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. This way, an examinee that mastered all required skills for an item 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, $\mathbf{Q}$ and $\neg\mathbf{S}$

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

Later in 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. Koren et al. (2009) for a brief description of some of the adaptation of these techniques in recommender systems.

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

The other skills assessment models we consider are based on what is referred to as Deterministic Input Noisy And/Or (DINO/DINA) Junker and Sijtsma (2001). 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 (Robitzsch et al., 2012).

2.6 Recent improvements

The previous sections of this chapter introduced skills assessment techniques to obtain the predictive performance given a dataset. Recall that these models are used for the purpose of defining a performance space used for assessing model fit, as briefly described earlier and detailed in chapter 3. Let us add to the description of the models a few recent developments that concern how the Q-matrix is determined.

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 Desmarais (2012) that was published in ITS conference. This article aims to find a method to derive these matrices for different types of Q-matrices. Finding the number of latent skills (i.e. the common dimension between matrices **Q** and **S**) is another important task that is described in section 2.6.1 . The text of section 2.6.2 is mainly borrowed form Beheshti et al. (2012) 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 Desmarais et al. (2014) 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 be derived from data (Winters, 2006; Desmarais, 2011). Winters (2006) 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. These studies show that only highly distinct topics such as mathematics and French can NMF yield a perfect mapping for real data.

Desmarais (2012) 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, **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.

2.6.2 Finding the number of latent skills

A major issue with Q-matrices is determining in advance what is the correct number of skills. This issue is present for both expert defined Q-matrices and for data derived ones as well.

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 (Beheshti et al., 2012). 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 introduced 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 Guyon and Elisseeff (2003)). 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 number of latent factors (skills) 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 on the real data.

2.6.3 The refinement of a Q-matrix

Very often, experts define the Q-matrix because they have a clear idea of what skills are relevant and of how they should be taught.

Validating of the expert defined Q-matrix has been the focus of recent developments in the field of educational data mining in recent years (De La Torre, 2008; Chiu, 2013; Barnes, 2010; Loye et al., 2011; Desmarais and Naceur, 2013). Desmarais et al. (2014) 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 (see (Desmarais and Naceur, 2013) for more details on ALS technique).

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.

2.7 Model selection and goodness of fit

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. Model selection in EDM is the task of selecting a statistical student model for a given data from a set of candidate models that are the best representatives of the data. Note that there could be a pre-processing step on the data itself to be well-suited to the problem of model selection but our study goes beyond that. The best fit is the model that is most likely to have generated the data. Selection is most often based on a model's "goodness of fit". The simplest way is to choose the best performer model. Models with higher predictive accuracy yield more useful predictions and are more likely to provide an accurate description of the ground truth.

On one 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 different measures, namely the best known is likelihood ratio. There exists different approaches to assess model fit based on the measure of goodness of fit. Below we describe few measures for goodness of fit that are commonly used:

2.7.1 Measures for goodness of fit

To find the prediction quality of each skills assessment model some metrics are used. There is a wide range of choices of metrics to evaluate a model and choosing an appropriate one is very important since usually candidate models are preforming with small differences and a good metric can highlight the benefits of one model vs. others.

There are different measures to represent the goodness of fit and this is usually either the sums of squared error (SSE) or maximum likelihood. Dhanani et al. (2014) in a survey compared three error metrics for learning model parameters in Bayesian Knowledge Tracing(BKT) framework. In their methodology they calculate the correlation between the error metrics to predict the BKT model parameters and the euclidean distance to the ground truth. These error metrics have been widely used in model selection researches. Below we will describe these metrics briefly:

— 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 given an observed outcome. This allows us to fit many different types of model parameters. This measure is mostly for estimating parameters and in next sections of this chapter we will see its application in model selection. Since in our study the student test result follows a binomial distribution then the likelihood can be defined as equation 2.8.

$$Likelihood(data) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)} \tag{2.8}$$

where p_i and y_i are the estimated and actual values of i^{th} datapoint. Applying natural logarithm on the right side of equation 2.8 will results log-likelihood. Hence it becomes more convenient in maximum likelihood estimation because logarithm is a monotonically increasing function.

— Sum of squared errors of predictions (SSE) is the other measure which is the total deviation of the response values from the predicted values as represented in equation 2.9

$$SSE(data) = \sum_{i=1}^{n}(y_i - p_i)^2 \tag{2.9}$$

A more informative measure is RMSE which the squared root of the mean squared errors:

$$RMSE(data) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2} \tag{2.10}$$

— There is another category of metrics that is widely used for classification purposes and we also use in our experiments to compare the ground truth prediction of two model selection approaches for a given data. These metrics rely on the confusion table which allows us to calculate the accuracy, recall, precision and F-measure values. A confusion matrix is a table that shows the performance of a classification method. A model selection method can also be tested as a classification method as we will show in chapter 3.2.4. Below a confusion table is presented where each row represents the number of instances in the actual class and each column represents the instances in the predicted class:

		Prediction outcome	
		Positive	Negative
Actual value	Positive	TP	FN
	Negative	FP	TN

where Negative in the context of EDM is a failure and Positive is a success to an item. There

exists four metrics that are:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$F-measure_{\beta} = (1+\beta^2).\frac{Precision.Recall}{\beta^2.Precision+Recall}$$

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 (Hastie et al., 2005). 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.

The other measure that we used in our study is calculating the distance to the ground truth. Assuming a model performance space where the performance related to each model belongs to a specific subspace, thus the closest neighbor to the performance of a given data is the model fit and the correlated distance is the measure for the goodness of fit. Details of this measure is given in chapter 3.

2.8 Related works

The following sections summarize two recent works (Desmarais and Pelczer, 2010; Rosenberg-Kima and Pardos, 2015) to assess a model fit which both rely on synthetic data:

2.8.1 On the faithfulness of simulated student performance data

Desmarais and Pelczer (2010) 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 (Vomlel, 2004) and De La Torre (De La Torre, 2008) 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 this thesis we are using this method as a part of our methodology.
— Generating simulated data: This is the method that Desmarais (Desmarais and Pelczer, 2010) used in his work. They used a set of predefined parameters to generated a result matrix based on a specific model.

Simulated data models

In (Desmarais and Pelczer, 2010) 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.

Methodology

Once the simulated data is generated base on the four models Desmarais and Pelczer (2010) 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.

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

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.

As we will see later, 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 (Desmarais and Pelczer, 2010). Obviously one of the reasons that synthetic data with IRT ground truth shows a good similarity with real data performance in their research is that the performances are over POKS model which shows closest performance to POKS model (The details will be discussed later in 5) oppose to other models that are linear and multi-skills.

To summarize the difference between our experiment and (Desmarais and Pelczer, 2010) 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. Desmarais and Pelczer (2010) 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.

2.8.2 Simulated data to reveal the proximity of a model to reality

The next recent work (Rosenberg-Kima and Pardos, 2015) is about distinguishing between a synthetic data and a real data. This work is an extension to their previous work (Rosenberg-Kima and Pardos, 2014) 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 (Rosenberg-Kima and Pardos, 2015) 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.

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 3
PERFORMANCE SIGNATURE APPROACH

This chapter presents the method we introduce to determine the model that is closest to the ground truth for a given data set. We refer to this method as "Performance Signature" approach, or "Signature" approach for short.

The basic principles of the method is to define a vector space of model performances, namely the models described in chapter 2. Then, synthetic data is created with each model, and each model's performances are measured over each of the synthetic data set. These measures represent a point in that space which is called **Performance prototype**. The synthetic data point associated with that point represents the ground truth prototype, or the centroid for classification. Given a new data set, a **Target Performance Vector** can be obtained. Then, we use a nearest neighbor approach with a correlation distance to identify the closest ground truth model.

Let us rephrase the method in more specific terms and consider a set of models, $\mathscr{M}$, and a vector **p** of length $|\mathscr{M}|$ that contains the performance of each model over a given data set. This **Target Performance Vector** represents a point in the performance space. For each model $M \in \mathscr{M}$, we determine a **Performance prototype** point in the performance space that corresponds to synthetic data generated with model M. Then, for a given data set, we find the nearest synthetic data set point, using correlation as a distance, and consider the model behind it to be the ground truth.

The rest of the chapter goes into more details about the method.

3.1 Model fit in a vector space framework

The performance data in table 3.1 will serve here to explain the method with a concrete example. In this table, 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* (section 2.4.2) and a seventh data set named *random* (random results that constrained to reflect the target data set parameters such as item and student success rate distribution) are added for comparison purpose.

As we can expect, the diagonal (in bold face, except for one, corresponding to the match between the underlying synthetic model and the model performance) generally 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

1. Note that the Synthetic datasets are generated based on different models. Namely, IRT means that the ground truth of the generated dataset assumed to be IRT model

Table 3.1 Vector space of accuracy performances

Model	**Synthetic data set** [1]						
	Random	POKS	IRT	DINA	DINO	Linear .Conj	Linear .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	0.97	0.58
NMF.Comp	0.75	0.76	0.79	0.59	0.93	0.70	**0.98**

behind the ground truth. However, this is not always the case. In this particular example, the DINA model show a better performance over the Linear Conj. data set, whereas this data set's corresponding model is the NMF Conj.

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 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 turn, 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 3.1 suggests that aligning the model with the ground truth does yield the best performance except for one case, but we will show more 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.

3.2 Research questions

Let us get back to the research questions and design implementation of experiments to answer them, This section explains a general framework for each experiment, later in chapter 5 the results of these

experiment with more details are given. To make it straightforward we break the contribution into four experiments:

1. What is the *performance vector* of student skills assessment models over real and over synthetic data created using the same models?
   — Experiment 1: Predictive performance of models over real and synthetic data sets: In a first experiment, we focus on showing the performance of all models described in chapter 2 over synthetic and real data sets. It provides an overview of the predictive performance of each model across the different synthetic and real datasets. The output of this experiment is the **Performance vector** or the **Performance signature** of a dataset.
2. Is the *performance vector* unique to each synthetic data type (data from the same ground truth model)?
   — Experiment 2: Sensitivity of the signature over different data specific parameters: This section tests the uniqueness of the Performance signature across different data generation parameters.
3. Can the *performance vector* be used to define a method to reliably identify the ground truth behind the synthetic data?
   — Experiment 3: Model selection based on *performance vector* classification: This part proposes a framework for model selection as well as a measure that represents a measure for the goodness of fit of a model by comparing the target *performance vector* of a real dataset and the performance prototype vectors obtained from each synthetic datasets.
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?
   — Experiment 4: Generality of the signature approach under different assumptions about the data: In a last experiment, we test the generality of the approach by classifying data sets with different data specific parameters in the *performance vector* space. To validate the approach, we need to rely on synthetic data for which we know the underlying ground truth model. A matrix is created with data sets generated from the different models, and each model performance is measured through a cross validation process (using experiment 1). This matrix allows us to classify a data set of unknown ground truth according to a nearest neighbor approach (using experiment 3). In fact, this experiment tests if the signature approach remains reliable on different conditions of parameters. And also it compares the accuracy of the classification between signature approach and the best performer approach.

3.2.1 Experiment 1: Predictive performance of models over real and synthetic data sets

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 3.1. 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.

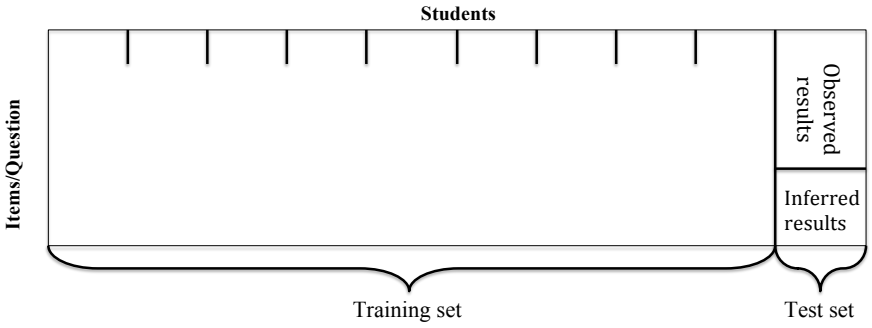


Figure 3.1 Data breakdown of cross validation process

For each dataset there exists a training set that contains 9 folds and a test set which represents a single fold. 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 to be used in the test set.

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 model specific parameters for assessing the model performance is presented in table 3.2 [2].

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 later in chapter 5. The R package `ltm` is used for parameter and skills estimation for IRT model and the R package `CDM` and `NMF` for Deterministic Input Noisy and NMF models.

[This is a description of experiments whose results are presented one chapter later. Not ideal, but I understand it is

2. Details of all parameters are given in chapter 4

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 • Partial order structure		

Table 3.2 Parameters of the predictive performance framework

also necessary to introduce the data generation chapter. We may leave it as is, but I add this note for the moment]

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

In this experiment we want to examine the effect of data specific parameters on the stability of the performance prototype vector in the performance space. We run the same experiment as before but with different contextual data generation parameters such as average success rate, sample size, number of latent skills, number of items, student and item score variance. This experiment can answer the question whether the pattern of performance signatures for datasets with the same ground truth remain stable across different conditions. Also it can answer whether the synthetic datasets to assess model fit should follow the same data specific parameters as the real data.

3.2.3 Experiment 3: Model selection based on *performance vector* classification

This experiment essentially introduces the "signature" approach. By the assumption that previous experiment proves that the uniqueness characteristic of synthetic data signatures makes it easy to identify the ground truth and also there exists some similarity between the pattern of performance signature of synthetic and real datasets (chapter 5 explains these results). Hence we can define a measure for the similarity of a synthetic generated dataset and a real one given a set of candidate models. Given that our objective is to determine the ground truth of a given data set, we will borrow data specific parameters from this data set to generate the synthetic data. Further details on data generation are given in chapter 4.

As explained before, signature approach search for the nearest neighbor of the target *performance vector* among the performance prototypes. The principal of this search is to find the closest performance prototype to the target *performance vector* in the performance space. Since this search is in a hyper space we used a linear discriminant approach which is the nearest neighbor classifier. This approach essentially splits the space into parts with a defined surfaces. However there exists other approaches to define a surface such as SVM (Cortes and Vapnik, 1995), single layer neural network (Lippmann, 1987) or LDA (Blei et al., 2003). We do not directly define any surface in the performance space but we just measure the closest neighbor. In this experiment we used Pearson correlation coefficient as a measure of similarity to find the nearest neighbor.

There are two approaches to search for the nearest neighbor in this space:

— Centroid performance prototype: In total there are 7 neighbors(points in performance space) for this search where each of them is a representative of a performance prototype of a skills assessment ground truth (Chapter 2 explains these models in details). Each centroid performance prototype is defined as the average of *performance vectors* of 10 datasets generated with the same model and data specific parameters but different seeding points. The nearest neighbor among these 7 centroid *performance vectors* becomes the estimated ground truth.

— Majority voting among 10 nearest neighbors: This approach considers all neighbors in the search without defining a centroid point for each performance prototype. In this case the distance between each neighbor's *performance vector* and the target *performance vector* is calculated and a majority voting among the ground truth of 10 nearest neighbors defines the estimated ground truth.

In this experiment we used the first approach to define the closest neighbor. However, both approaches show identical results.

3.2.4 Experiment 4: Generality of the signature approach under different assumptions about the data

The very last experiment is to test the generality of signature approach to assess a model fit over different data specific parameters. Despite the fact that different data specific parameters may affect the performance signature (as we will see in the results of experiment 2), this experiment tests the reliability of this approach to estimation of the ground truth on different assumptions about the data. Akin to experiment 2, this experiments relies on synthetic datasets since the ground truth and data specific parameters are known.

Considering 6 data specific parameters which are presented in table 3.2 and 4 different values for each parameter, while others have default values, results in 24 different conditions for the data generation process. The goal of this experiment is to obtain the accuracy of signature approach to

classify any dataset with each of these 24 conditions. Therefore, with each condition we generate 10 datasets with different seeding point for each skills assessment ground truths. This results in 60 datasets for each condition. let us name this set of 60 datasets as a **group**. Totally 24 **groups** can be generated.

The classification is preforming among the members of a **group** where all datasets share the same data specific parameters. We randomly select a dataset whose *performance vector* becomes the target *performance vector*. Consequently the other 59 datasets in the **group** are neighbors with performance prototype vectors. As explained in experiment 3 we can calculate the distance between the target *performance vector* and each of neighbors for the purpose of model selection. Finally a majority voting among the ground truth of 10 nearest neighbors defines the ground truth. This process repeats for each **group**.

Obviously the highest performance in a *performance vector* is the classification result for the "best performer" approach. Later in section5.5 we use the classification measures from section 2.7.1 to compare the reliability of these two approaches.

CHAPTER 4
SYNTHETIC DATA GENERATION

The framework of this study relies on synthetic data. The performance signature vectors are based on synthetic data generated from each model of the performance space. Therefore, we dedicate a full chapter to the process of generating synthetic data from each skills assessment models described in chapter 2.

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: Models have their own parameters, some of which may be shared among models, such as a Q-matrix for multiple skills models, some of which are more unique such as the item discrimination parameter of the IRT class of models. 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): The data specific parameters are common to all data sets. They have to be predefined, but they may be set to reflect a data set for which we want to find the ground thruth. For example, the item difficulty distribution or the student success rate distribution are two data specific parameters. Data specific parameters can also be a model parameter, such as item difficulty and discrimination.

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

4.1.1 Assessing parameters

Both types of parameters are required to generate synthetic dataset. There are few means to obtain these parameters:

— **Parameters estimated form real data**: In general, models define means to estimate their

Skills Model			Parameters: Model specific		Parameters: Data specific
Contributed skills	Multiple	NMF Conj.		• Q-matrix • Students skills mastery matrix	• Number of students • Number of items
		NMF Add.			
		DINA	• Slip • Guess		
		DINO			
	Single	IRT	• Student ability • Item difficulty • Item discrimination		• Number of skills • Test success rate • Student score Variance • Item score Variance
		Expected	• Student Odds • Item Odds		
	Zero	POKS	• Initial Odds • Odds ratio • Partial Order Knowledge Structure (KS) (includes two alpha error parameters for its induction from data)		

Table 4.1 Parameters involved in synthetic data generation

parameters from data. Some are trivial to estimate, such as student success rate distribution if we assume a Gaussian distribution, for eg., and some are much more complex, which is the case of latent factors such as the number of skills and the Q-matrix itself.

— **Initialized from a random distribution**: In the case that there exists no reference dataset to derive parameters we can randomly select some samples based on a distribution that can be estimated form the data. Some parameters have specific conditions that if they become violated it changes the definition of the parameter. More details is given in next sections.

— **Expert given and arbitrary parameters**: Latent factors are often not derived from data but arbitrarily defined by expert or by some educated choice. The Q-matrix is the most obvious of such parameter, but some are the guess and slip parameters which can prove difficult to correctly estimate from data.

These means can be combined. For eg. guess and slip can be estimated based on maximum likelihood of a given model over existing data, and an expert can provide an estimate of the variance of the guess and slip on a per item basis, which will eventually be the basis of a random generation of the guess and slip parameters of an item pool, say from a Gaussian distribution.

The rest of this chapter describes the data generation process based on each skills assessment technique in chapter 2.

4.2 POKS

As mentioned before this technique does not directly model latent skills and the inference relies on observable items only.

4.2.1 Obtaining parameters

This model has three parameters with which the inference in possible: Partial order knowledge structure (let us name it KS for shot), Initial probabilities, Odds ratio. To extract these parameters from an existing data we use the POKS [c0] R package to learn all the parameters given a dataset. Sometimes these parameters are defined by an expert. In the absence of these two alternatives, these parameters should be generated randomly. Below we describe how these parameters are generated with different conditions and constraints.

Partial Order Knowledge Structure (KS)

This parameter is in fact a directed graph that shows the relation and the dependency among a set of items. It reflects the order of learning a set of items in the same problem domain. Items are "nodes" in the graphical structure and the "parent-child" relation represents a prerequesite relation. Symmetric links are allowed and represent items that are similar. They are the only type of cycles possible. Also in the current version of POKS transitive relations between items are explicitly ignored.

KS is a complex parameter which has two other parameters involved in itself:

- Number of independent graph components: If graphiof a KS is not a connected graph, the graph is said to be composed of independent graph components, each of which is connected. This is an important parameter as it represents the number of indepent knowledge topics.
- 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 a 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. There are still some other parameters that could change the test success rate and item score variance. In this sense manipulating one parameter may affect others.

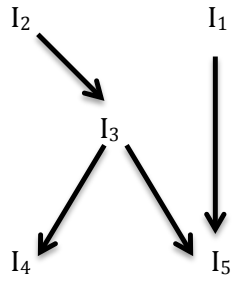
[These parameters come out of nowhere...] In the current study, the two parameters, α_p and α_c, are respectively set at 0.85 and 0.10. Other input parameters can make the random generation more specific, such as the number of links and number of independent trees in a KS. These parameters will change the item variance in the test result matrix.

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

Adjacency matrix is one way to represent the KS. To avoid having a cycle in the structure we randomly assign 0 - 1 values to the 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 the cells of the upper triangular of this matrix. Figure 4.1 shows a adjacency matrix associated with its graphical structure.

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

(a) Adjacency matrix



(b) Graphical representation

Figure 4.1 An example of random KS with 5 items

Once the KS is generated we should initialize the two other parameters : initial probability of all items and odds ratio for some pairs of items. One of the important concepts for this purpose 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 4.1 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. This is an example where a data specific parameter is also a model parameter. In general if node X_a implies node X_b ($X_a \rightarrow X_b$) then node X_b should have a higher initial probability of success because in this relation item X_a is the harder problem. The allocation of initial values to items should be done with 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's parents. For a node that is a root r_{max} is 0. Obviously it 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 the least level.

As mentioned before, the initial probability of success for items come from these values. To sample test outcome for a record(student) we start with initial probabilities. This set of values changes when a sample is assigned to each item. Let us call this variable success probability set as 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 sampling process for a single record preforms item by item and in each step S vector gets updated. The details of updating the "state of odds" and assigning samples to items are 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. We also 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 follow the same steps to generate data based on POKS model.

There are two types of odds ratio in the POKS model. Consider I_3 in figure 4.1 which has two children and a parent . If the sampling result for this item (with respect to its initial odds) becomes 1 (success) then we 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 "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 (failure) then we 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 ratio of the item H given evidence E_i.

In our experiments, in cases of random parameter generation, we assign values greater than 0.5 to

"true odds ratios" and smaller than 0.5 to "false odds ratios" for those pairs of items that have link in KS.

4.2.2 Data generation

Previous sections explained the process of obtaining a random generated set of parameters for POKS. This is useful if the expert defined parameters or even the estimated parameters from real data are not available. In this section we explain the process for sampling data points values to create a synthetic student test result matrix given a set of parameters. Each record in this approach represents 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 odds 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 propagates through a breadth-first search.

Algorithm 1 POKS data generation

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

4.2.3 Data specific parameters

In some experiments, we need to generate a synthetic data with specific mean success rate for both students and items, and with specific variance, or specific entropy.

One 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. To control the student scores variance, one can scale the initial odds of each record such that the distribution of the initial odds stays the same for all students; 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 success rate, the initial odds can be scaled regardless of the student 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

The standard IRT models student response outcome with a single skill. But in addition to the skill mastery, IRT includes three other parameters. Akin to all models these parameters can be derived from real data or an expert can define them, it is also possible to initialize these three parameters with a distribution. The rest of this section describes the steps to generate these parameters and the synthetic data.

4.3.1 Generating parameters randomly

Equation 4.3 shows the probability of a student with ability θ to succeed item j which has the difficulty of b_j and discrimination of a_j based on IRT-2PL model.

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

We assume the following distributions for this model parameters:

$$\theta \sim \mathcal{N}(\mu, s_t)$$
$$a_j \sim \text{Pois}(\lambda_a)$$
$$b_j \sim \text{Pois}(\lambda_b)$$

The student related parameter is θ and it follows a standard normal distribution with two parameters μ and s_t which are respectively mean of student ability and individual examinee standard deviation. These two parameters can be obtained from an existing dataset or they can be assigned to specific values if desired. Two parameters a and b are item related parameters that are generated based on a poisson distribution with λ_a and λ_b to control item score variance.

Extremely large or small values for each of these parameters will result in an unrealistic outcome generation. Hence we bounded values for these parameters:

$$-4 < \theta_i < 4$$
$$0.5 < a_j < 3$$
$$-3 < b_j < 3$$

where i and j are students and items respectively.

4.3.2 IRT synthetic data process

Once all parameters are obtained we can generate the simulated data with equation 4.3. The results of 4.3 are probabilities between $0-1$ that should be discretized to 0 or 1 values by rounding. Test score success rate is the threshold to discrete the test result probabilities. Algorithm 2 shows the steps to generate IRT based synthetic data. θ and b are also a kind of data specific parameter which can control the item and student variance in the generated data.

Algorithm 2 IRT data generation

1: $a = \text{Pois}(\lambda_a, \text{Number of items})$
2: $b = \text{Pois}(\lambda_b, \text{Number of items})$
3: $\theta = \mathcal{N}(\mu, s_t, \text{Number of students})$
4: **for** each record $i \in m$ (number of students) **do**
5: **for** each item $j \in n$ (number of items) **do**
6: $R_{ij} = (\frac{1}{1+e^{-a_j(\theta_i - b_j)}} > \text{Test score successRate})$
7: **end for**
8: **end for**

4.4 Linear models

Generating synthetic data with linear models is one way to cover multiple skills. In this study, Q-matrix is the means to represent the mapping between skills and items. If the Q-matrix needs to be inferred from a data set, NMF is the technique we use in this study. Depending on the type of a Q-matrix, different models could be applied to NMF technique. In this section we describe how to generate synthetic datasets that reflect the characteristics of each model of the Q-matrices used for factorization technique. Section 2.5 described all details to learn parameters of these models given

a dataset. This section discusses the process of generating synthetic data given a set of parameters for these models.

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. As before these parameters can be inferred form an existing dataset but many existing datasets offer few expert defined Q-matrices. Some datasets even have more than a single expert defined Q-matrix with different number of skills. Section 2.6.3 introduced methods to refine an expert-defined Q-matrix. A Q-matrix can also be generated randomly. As mentioned, there are three types of Q-matrices. In this study we use conjunctive and additive model in the factorization technique.The following sections describe generating parameters for each model.

4.4.1 Q-matrix

Q-matrix is a required parameter for all linear models. Few mandatory parameters are required to create a random Q-matrix:

Parameters

- Number of items
- Number of skills: In some context there is a constraint for the number of latent skills which is: $k < nm/(n+m)$ (Lee et al., 1999) where k , n and m are number of skills, students and items respectively.
- Maximum number of skills per item: This parameter can also reflect item difficulty level. The difficulty of the two-skills items will further increase by the fact that they require the participation of their skills.
- Item score variance: The only way to apply item score variance is to manipulate the Q-matrix. Harder items should require more skills than easier ones. Therefore the ratio of contributed skills to the total number of skills shows the variance of item scores which will be reflected in the final result 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 $\sum_{k=1}^{Max} \binom{n}{k}$ candidate items span this space of combinations. For example 21 items for 6 skills and a maximum of 2 skills per item. This matrix is shown in Figure 4.3(a). Note that red cells represent 0 or missing skills and yellow cells are 1 or presence of skills. Items 1 to 15 are two-skills and items 16 to 21 are single-skill. All skills have same weight in a conjunctive type of Q-matrix. The Q-matrix can be created with randomly replicating or eliminating some candidate items to adjust the number of items to the desired number. Also item score variance can be controlled by

picking some appropriate items to create high or low variance.

Additive vs. Conjunctive

The proposed approach to create conjunctive type of Q-matrix differs from additive type where a normalization process changes the values of skills. The normalization process forces the weights that are assigned to skills to sum up to 1 for each item. Figure 4.2(a) shows an additive type of Q-matrix where skills have same weights for each item but different weights among items.

4.4.2 Skills mastery matrix (student profile)

Skills mastery or student profile is the other mandatory element for all linear models in this context, since the synthetic data is the product of the Q-matrix by the skills mastery matrix. Mapping skills to students also requires some mandatory parameters to create more specific synthetic test result matrix:

Parameters

- Number of students
- Number of skills: This number should match with the one in the Q-matrix. In this matrix skills show the ability of students to answer items.
- Student score variance: In order to apply student score variance one should consider different abilities for students. The ability is, in fact, reflected by the set of skills that an individual is mastered. The assigned ability level will be reflected in the generated test result. Both Item and student score variance in our study are taken from a beta distribution:

$$\text{Student score} \sim Beta(\alpha_S, \beta_S)$$
$$\text{Item score} \sim Beta(\alpha_I, \beta_I)$$

  where α_S, β_S, α_I and β_I are the shape parameters of the $Beta$ distribution of student and item score.

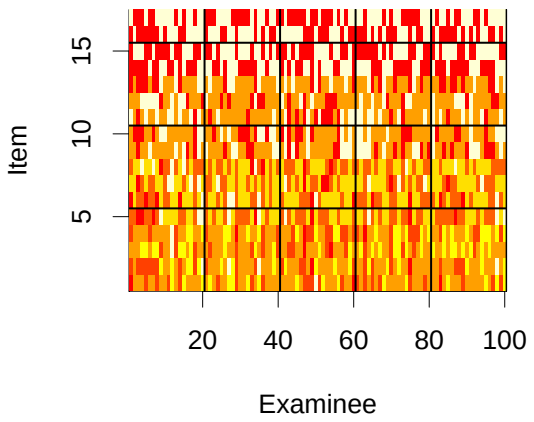
Students with fewer mastered skills should have low test score but in reality skills have different weights. For example examinees E_1 and E_2 are mastered in two unrelated skills S_1 and S_2 respectively. Although they both master single skill but they do not necessarily get same test scores. In this thesis we consider that skills have same level of difficulty which means that they have same weight in both Q-matrix for each item and skills mastery matrix for each student.

items

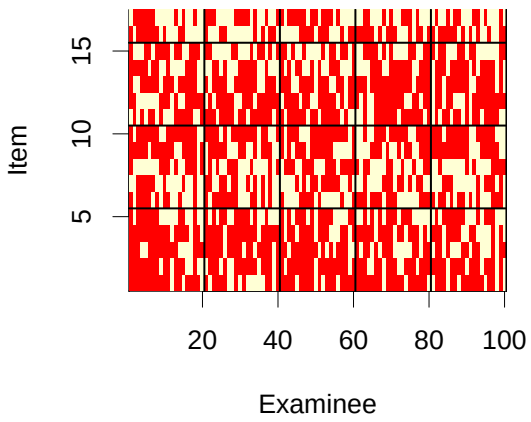
skills

0.00	0.00	0.25	0.00	0.00	0.00	0.33	0.33	0.5	0.0	0.0	0.5	0.0	0	0	0	0
0.25	0.00	0.25	0.25	0.00	0.00	0.00	0.33	0.0	0.0	0.0	0.0	0.0	0	1	0	0
0.25	0.25	0.25	0.25	0.00	0.33	0.00	0.33	0.0	0.5	0.0	0.0	0.0	1	0	0	0
0.25	0.25	0.00	0.00	0.33	0.33	0.00	0.00	0.5	0.0	0.5	0.0	0.0	0	0	0	0
0.25	0.25	0.00	0.25	0.33	0.33	0.33	0.00	0.0	0.0	0.0	0.0	0.5	0	0	1	0
0.00	0.25	0.25	0.25	0.33	0.00	0.33	0.00	0.0	0.5	0.5	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 4.2 Additive model of Q-matrix and Corresponding synthetic data

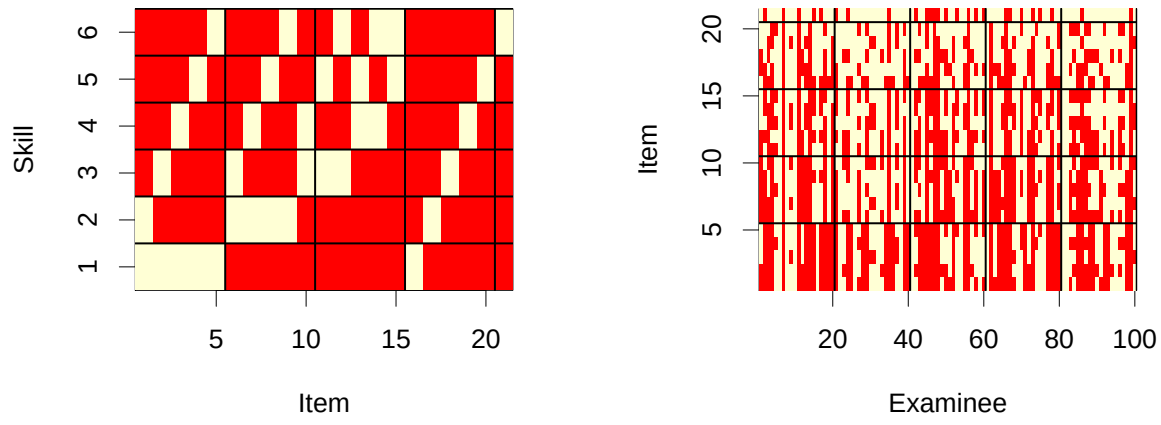
4.4.3 Synthetic data

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. In the additive model a simple cross product of the Q-matrix and skills mastery will generate the test result matrix. For conjunctive a negation operator should be applied on both skills mastery and test result matrices:

$$\begin{aligned} \text{Conjunctive model: } \mathbf{R} &= \neg\left(\mathbf{Q} \times (\neg\mathbf{S})\right) \\ \text{Additive model: } \mathbf{R} &= \mathbf{Q} \times \mathbf{S} \end{aligned} \tag{4.4}$$

Figure 4.2(c) shows an artificial result matrix based on additive model of Q-matrix . Since the model is additive, there are some pale cells and the paler a cell is, the more chance a student has to succeed the question. In conjunctive model the result matrix is either 0 or 1.

[Stopping here for now in this chapter. 2016.01.19. There are many typos, grammatical errors, and ill structured sentences...]



(a) Q-matrix of 6 skills

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

Figure 4.3 Q-matrix and an example of simulated data with this matrix. pale cells represent 1's and red ones represent 0's.

4.4.4 Noise factor

Finally, two more parameters are used in the simulated data, namely the *slip* and *guess* factors. They are essentially noise factors and there are three approaches to apply these factors to the result matrix:

- Student based: each student would have specific amount of these factors in his test result.
- Item based: this represent how tricky a question can be. The greater they are for an item, the more tricky that item would be. This approach is used in DINA/DINO models (next section describes these models).
- Overall: For synthetic data generated by linear models we use this type of noise. For example, with $s\%$ of slip and $g\%$ of guess factor ,this will result in approximately $s\%$ of the succeeded outcomes to become failed, and $g\%$ of the failed outcomes to become succeeded in the test result matrix.

Two samples of synthetic result matrices for conjunctive and additive model are given in figure 4.3(b) and 4.2(c) respectively where pale cells represent a value of 1 and red cells are 0. Examinee ability shows up as vertical patterns, whereas item 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. The same situation holds for additive model in figure 4.2, the patterns in figure 4.2(b) are more tangible where there are shades of colors that indicate different probability of success. Therefore, for a synthetic result matrix based on additive model, the values should be discretized with respect to an average success rate parameter. Figure 4.2(c) shows a Discretized version of result matrix with 20% slip and 10% guess factor.

4.5 Cognitive Diagnosis Models

From the family of cognitive diagnosis models we choose two models which are deterministic input noisy AND/OR. Detailed description of these models is given in section 2.5.3. This section discusses the approach that we took to generate synthetic datasets based on these models. The same as linear models, these models also represent the mapping of multiple skills to items and examinees in the form of a Q-matrix and skills mastery matrix respectively. To borrow the Q-matrix form an existing dataset we use NMF because these models can not infer a Q-matrix but they can estimate other model specific parameters such as student profile, slip and guess given a Q-matrix.

To obtain a conjunctive type of Q-matrix for DINA model we follow the same approach as described in section 4.4.1 where the normalization process does not apply. Generating the student profile is almost the same as section 4.4.2 but there are other parameters involved:

4.5.1 Skill space

Consider K skills for a cognitive diagnosis model, there exists 2^K combination of skills to be considered for a student profile. This shows the maximum capacity of skill space. One of the parameters of student profile is skill class where only some candidates from this combinations are allowed to be presented in skills mastery matrix. This parameter has a benefit that can control the prerequisite skill in a student skills mastery vector given a partial order structure of skills.

4.5.2 Skill distribution

The skill class set is associated with a distribution which defines the probability of appearance of each combination indicated in skills class. i.e. If there exists 3 skills then the skill space will be 8 combinations and assuming skill class consists of 4 combinations out of these 8, then skill distribution must be a vector of length 4, with sum equal to 1. Student score variance can also be set through this parameter. In this research, we use a normal distribution for this parameter.

$$Student\ skills \sim \mathcal{N}(\mu, s_s)$$

A skill mastery matrix can be generated once these parameters are set.

4.5.3 Slip and Guess

one of the characteristics of DINA/DINO models is that they incorporate slip and guess factors. These parameters are essentially noise factors and in these models they are treated item-wised (section 4.4.4). We defined a uniform distribution to cover a range of values for slip and guess:

$$Slip \sim U(S_{lower-bound}, S_{upper-bound})$$
$$Guess \sim U(G_{lower-bound}, G_{upper-bound})$$

Once all the parameters are prepared we can generate an outcome of test j for examinee i from equation 4.5 where ξ_{ij} is calculated based on AND or OR gates. 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. For DINO a disjunction is used 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.

$$R_{ij} = (1 - s_j)^{\xi_{ij}} g_j^{1-\xi_{ij}} \tag{4.5}$$

4.6 Educational data generator

Despite 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. Trinh (Trinh, 2015) created a package that generates synthetic data under a specific model's assumption given a set of parameters. As mentioned before these parameters can be fine grained which can be combined to create a more complex parameter. In this sense we can create a hierarchy for the complexity of these 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 synthetic data.
- Second: There exists no conflict between the given parameters. For example given "number of skills" as a simple parameter equal to 4 and a predefined "Q-matrix" as a complex one 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.

For some of parameters we can use default values but some are mandatory such as number of skills. Therefore based on the hierarchical structure among parameters of a model we can define sets of sufficient parameters to generate data based on a model. Due to this structure there are different combination of parameters that can generate a synthetic data. For example, a synthetic dataset based on DINA can be generated with {Q-matrix, Skills mastery matrix, Slip and Guess vectors} or {Number of skills, Items and students}. Figure 4.4 shows the hierarchical structure of the required parameters for linear conjunctive model. This graph can be extended to other models.

There exist some intermediate steps between the lower level(simple) and higher level(more complex) parameters which can potentially result in a variance in the generated data. The goal of creating this package is to efficiently use the available parameters to generate data based on different models where potential conflicts and exceptions are covered.

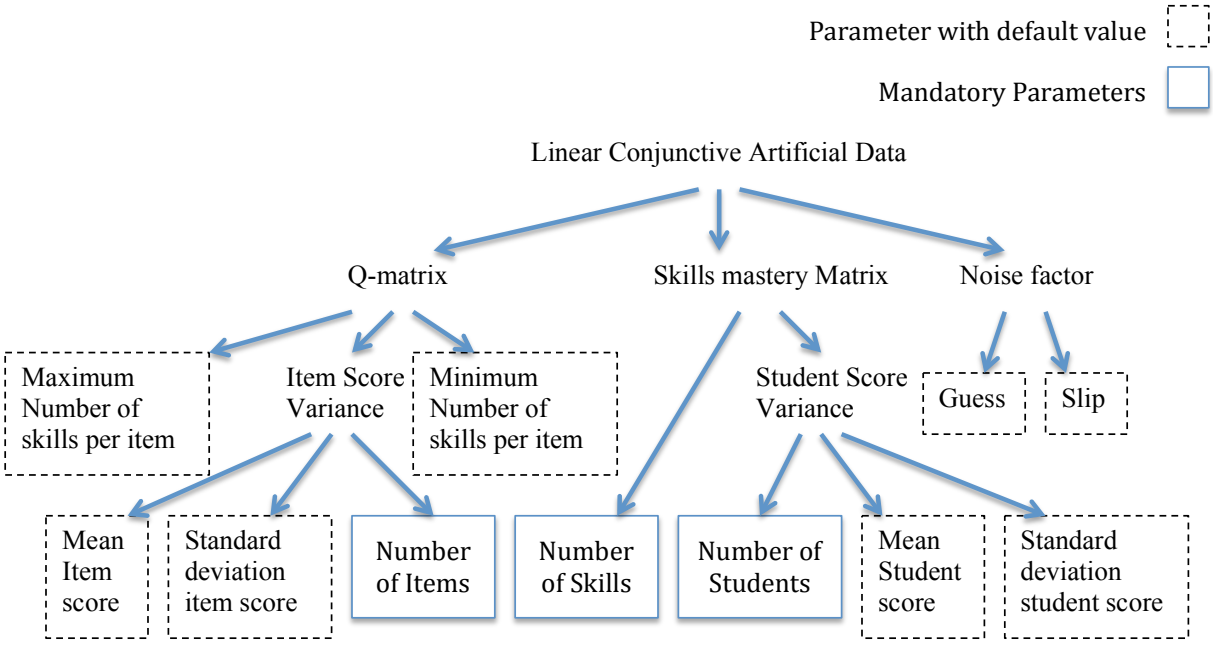


Figure 4.4 hierarchical structure of parameters of linear conjunctive model

CHAPTER 5
EXPERIMENTAL RESULTS

Chapter 3 introduced four experiments that each address a research questions of this thesis. In adddition to synthetic datasets, they require real datasets. In this chapter we describe real and synthetic datasets to examine the hypothesis and discuss the results of each experiment.

5.1 Data sets

In experiment 1 we want to extract the predictive *performance vector* of models over real and synthetic 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 5.1, 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 5.1 also reports the Q-matrix used for each dataset. As can be seen, some synthetic data sets share their Q-matrix with real data sets. This sharing allows greater similarity between the synthetic data and a real data counterpart that shares a Q-matrix. Other parameters used to create the synthetic data sets were also obtained from real data sets with the same intent of allowing better comparison. Chapter 4 shows details of generating each synthetic dataset.

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 (Tatsuoka et al., 1984). 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 (Vomlel, 2004) 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 (Robitzsch et al., 2012) and NPCD (`http://cran.r-project.org/web/packages/NPCD/`) R packages. The Q-matrices of the real data sets were made by experts.

Note that we use these datasets for experiment 3 which estimates a ground truth model for a dataset based on *performance vector* classification.

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 5.1 Datasets

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 synthetic datasets from linear and Cognitive Diagnosis models 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. For skills mastery we set α_I, β_I, α_S and β_S parameters equal to 1.5 to avoid high or low entropy for student and item variance. Skills space for DINA/DINO models considers all possible combination of skills with same probability of appearance in the skills mastery matrix. The synthetic datasets with linear model ground truths do not have any noise factor where *slip* and *guess* values of 0.1 and 0.2 respectively were set for synthetic data with DINA and DINO models.

Note that the first 3 models (Expected, IRT, POKS) do not rely on any Q-matrix neither for the data generation process nor for learning phase, but the DINO/DINA and matrix factorization assessment models still require one during the learning phase. For example a synthetic data with IRT model does not accompany any Q-matrix to preform experiment 1 which is assessing predictive *performance vector* of models but DINA requires one in the learning phase of this experiment. To define these Q-matrices (denoted $\mathbf{Q}_{0x}$ in table 5.1, a wrapper method was used to first determine the number of skills according to Beheshti et al. (2012), then a Q-matrix was derived with the deterministic ALS algorithm (see Desmarais and Naceur, 2013, for more details on ALS technique), starting with an initial random Q-matrix.

5.2 Results of Experiment 1: Predictive performance of models over real and synthetic datasets

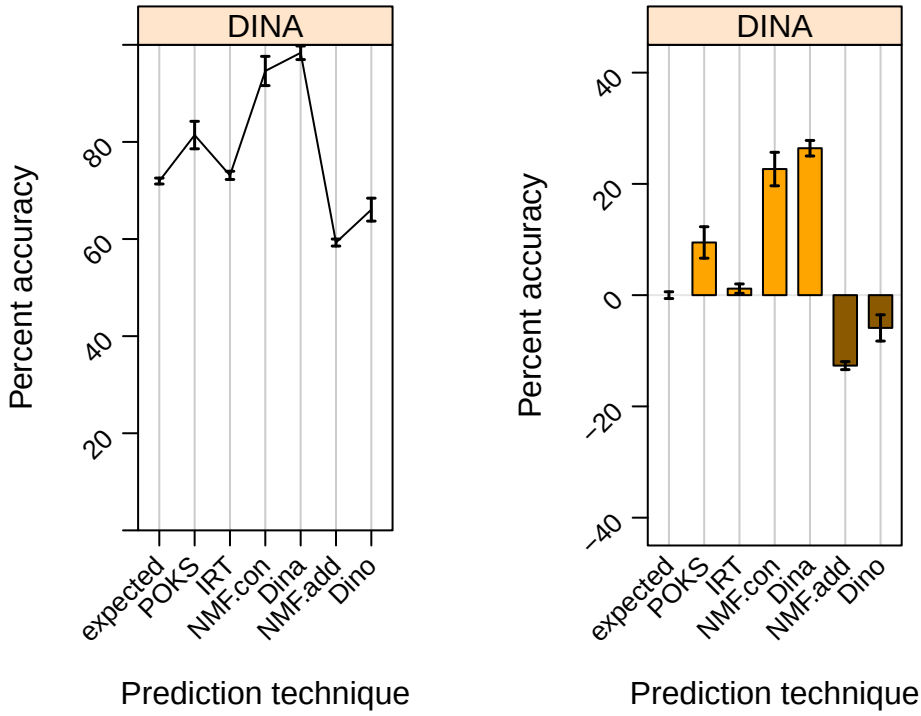
There are two possibilities for visualization of the predictive performance of seven models over a dataset, Figure 5.1 shows these two representations of the same results data. We used both of them in this thesis, whichever is deemed more appropriate to ease understanding. Figure 5.1(a) shows the signature performance in terms of percent accuracy of seven models over a dataset. Figure 5.1(b) centers the performances with respect to the accuracy of the *expected* model which serves as a baseline. When a model performs below that performance, it can be considered a relatively weak performer since the expected value is simply the combined student and item mean.

Figures 5.2 and 5.3 show the performance of each model over the synthetic and real data sets described in table 5.1. 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 the 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.

[If you have the scripts to quickly produce the figures, it would be worth using a 95% confidence interval which is the standard deviation divided by the square root of the number of instances (10). A 95% c.i. is 2 standard deviations and the sqrt of 10 is about 1/3, so the bars should be around 2/3 of the ones shown. If recreating these figures is too much work, we can simply mention this fact.]

5.2.1 Discussion

For the synthetic data sets, as expected, when the generative model behind the data set is the same as the skills assessment technique, the corresponding technique's performance is the best, or close



(a) Absolute accuracy percentage.

(b) Relative accuracy (centered at the *expected* model performance).

Figure 5.1 Two types of representation of predictive performance of 7 models over DINA generated dataset

to the best. And this performance is also always above the expected value performance, except for the random data set where no model can do any different than the expected value, which is what we would expect.

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 *performance vector* across models. The capacity of recognizing a data set's true model relies on this uniqueness characteristic.

For the real data sets, the *performance vector* among the techniques shows smaller discrepancies and is closer to the baseline.

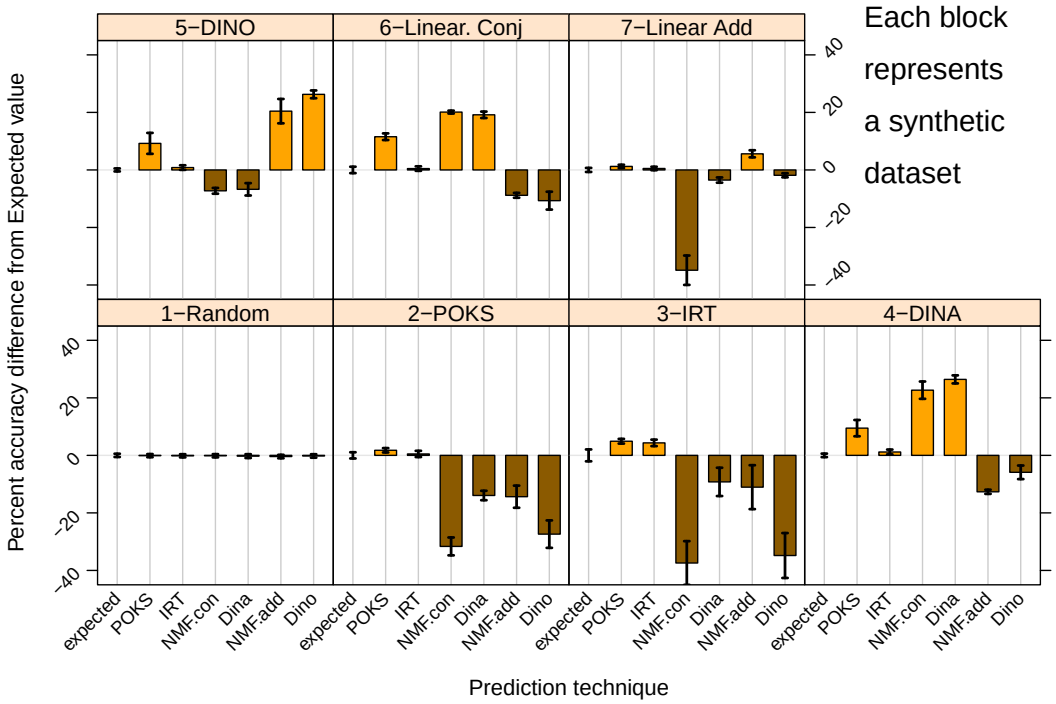


Figure 5.2 Item outcome prediction accuracy results of **synthetic data sets**

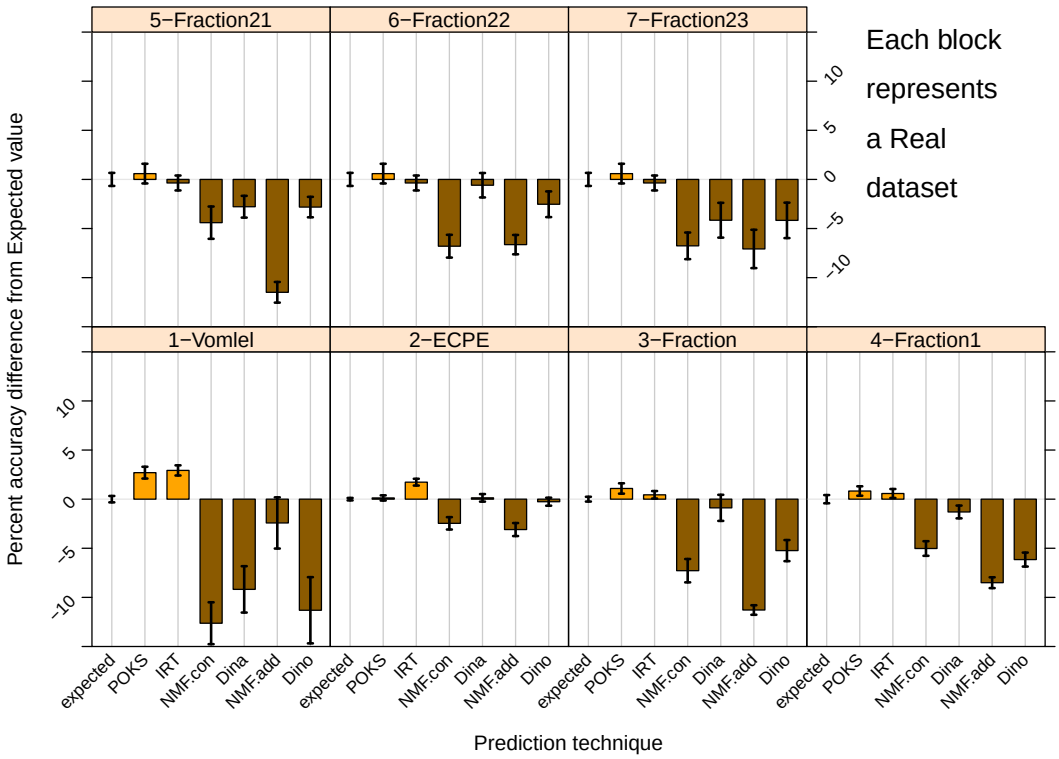


Figure 5.3 Item outcome prediction accuracy results of **real data sets**. Note that the y-scale is different than the synthetic data set.

An interesting finding is that in most cases, the best performer is close to the baseline. Moreover, a majority of models do worst than the baseline on most of data sets. This is particularly surprising for the DINA model which is a relatively well accepted model in psychometrics and student modeling. Even the highly used IRT model is not showing a better performance than the *expected* model. This is due in part because the estimated from *expected* model can rely on at least 19 observed items, but it is still indicative of a weakness of the models when we compare these results with the synthetic data results where the underlying model often reaches 20% above the *expected* baseline. Also noteworthy to notice is that for the POKS and IRT synthetic data, the highest performances are notably lower than for the other models. We will study in a later experiment the performances when fewer items are observed.

[It is surprising that POKS does not do any better, and even worst than Vomlel.]

[Where is the mention that the target item to predict is the last one? Or is it? It would be worthwhile to provide more details on this in this chapter or the previous.]

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 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 or even cognitive diagnosis models 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 amplitude and the differences found in the synthetic data sets models (the scale of figure 5.2 is about 4 times that of figure 5.3). One exception is the IRT 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. Details of this conclusion is given in the next section. [I do not get to the same observations here.]

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

The results of previous experiment show that the performance signature pattern is unique to each dataset among datasets with different ground truths. The next question to address is whether they

are stable in addition to be unique. This question motivated us to test the sensitivity of this pattern over wide span of data generation parameters. Therefore in this experiment we generated synthetic datasets with different conditions of the parameters that is presented in table 5.2. We should note that all other parameters are set to a default value while we are testing a specific parameter.

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

Akin to the previous experiment we assessed the predictive performance signature of these datasets on the basis of 10-folds cross-validation. We also 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.

5.3.1 Results and discussion

[Why not name "Bayesian generated" simply "POKS"? It is hart for the reader to make the link.]

Figure 5.4 shows the results of extracting performance signature from datasets with different number of students. Obviously, the signature pattern did not change significantly except for IRT-based generated dataset for small sample size [I do not see this??].

[Next paragraph: This is the number of items observed, right? Not the number of items in total. And what are/is the

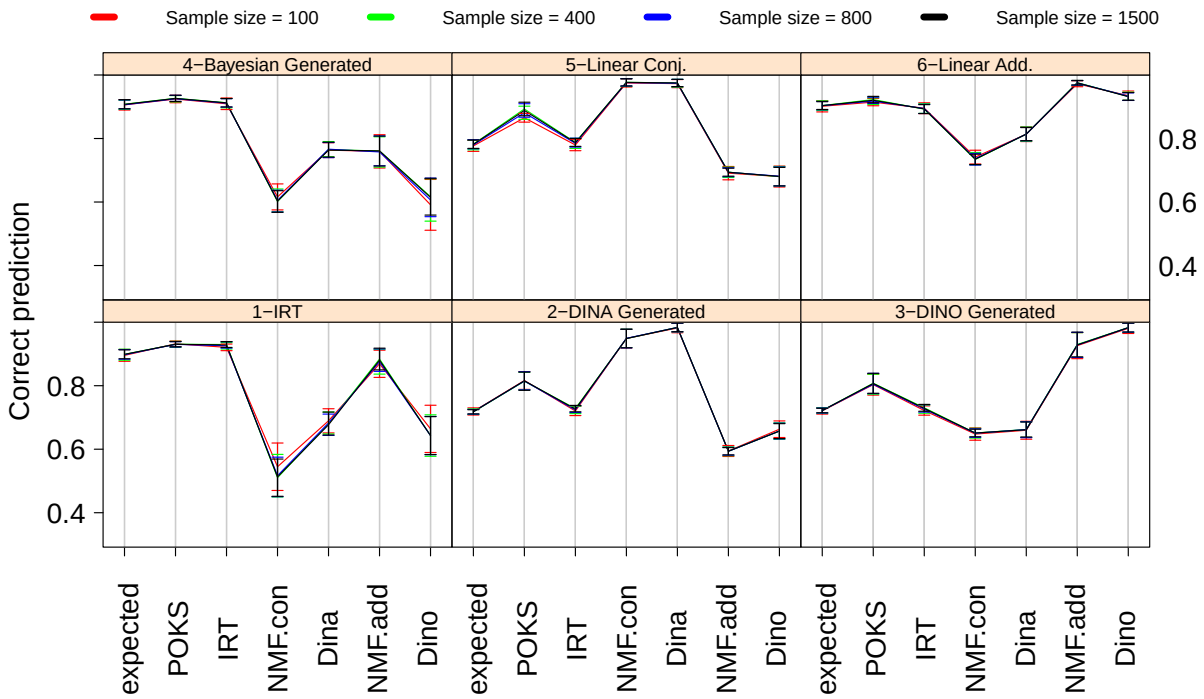


Figure 5.4 Variation of **Sample Size** Over synthetic data sets

target item/s? Is it still a single item, such as the last unobserved one? And if it is a single one, is it always the same one? chosen at random?]

The other parameter is the number of items. Figure 5.5 shows how the performance signature has been changed on different parameter assumptions. For three types of synthetic data (IRT, POKS, Linear additive) the pattern of the signature does not change across this parameter but for the others (Linear Conj. DINA, DINO) it shifts down once the number of items degrades. This result highlights the role of this parameter in the assessment of *performance vector*. It also shows that even for synthetic data the ground truth should not necessarily be close to 100%.

[Here again, some details are missing. What are the Q-matrices? How are the matrices with different number of skills obtained? Maybe in the discussion later, or here, questions such as "what if we have matrices with different number of items per skills, in particular one skill per item?]

The next parameter is the number of skills which is a latent parameter. Figure 5.6 shows the results of running experiment 1 on datasets with different number of latent skills. Synthetic data based on IRT and POKS can not be affected by this parameter because they are either zero or one skill models. However in the learning phase of these datasets this parameter is required. The signature pattern did not change substantially for these datasets except for linear conjunctive model. For other datasets the signature pattern stays stable while it slightly shifts upward on the correct prediction

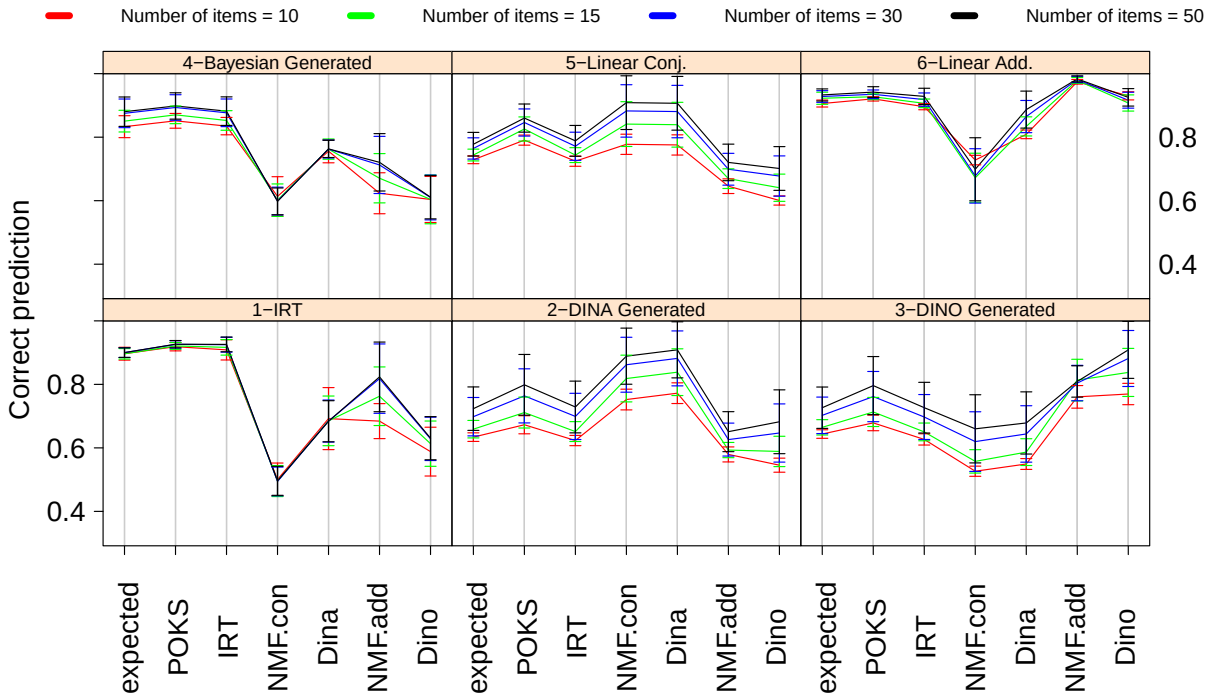


Figure 5.5 Variation of **Number of items** Over synthetic data sets

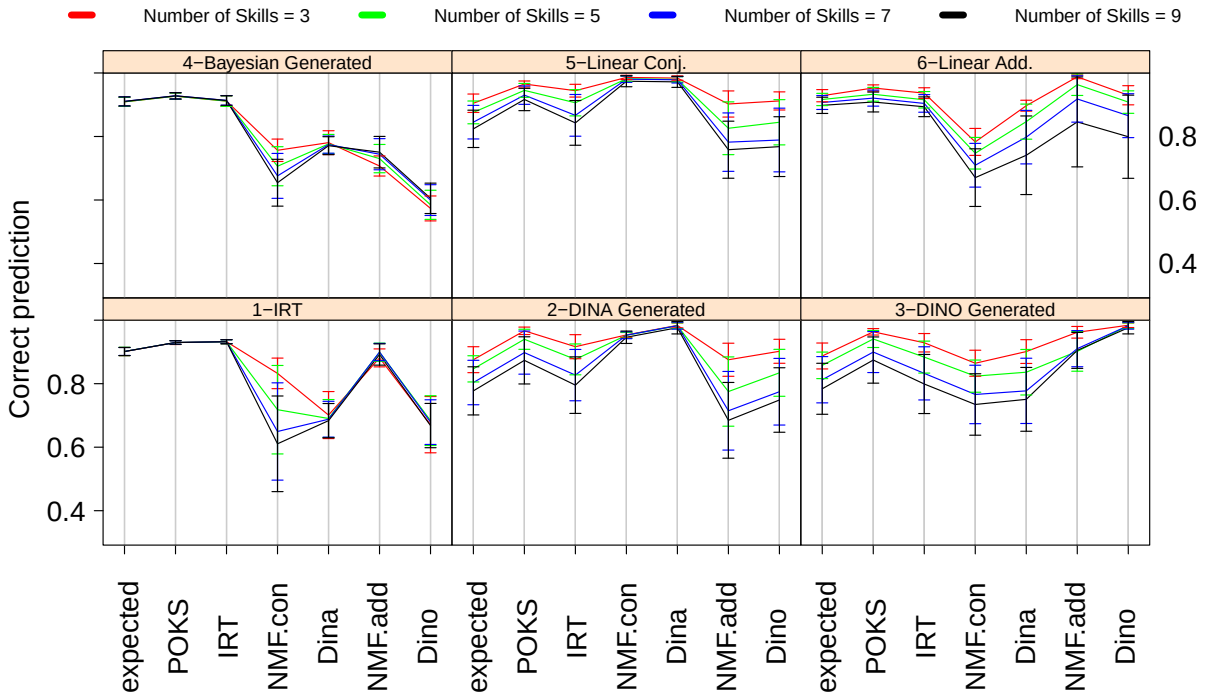


Figure 5.6 Variation of **Number of skills** Over synthetic data sets

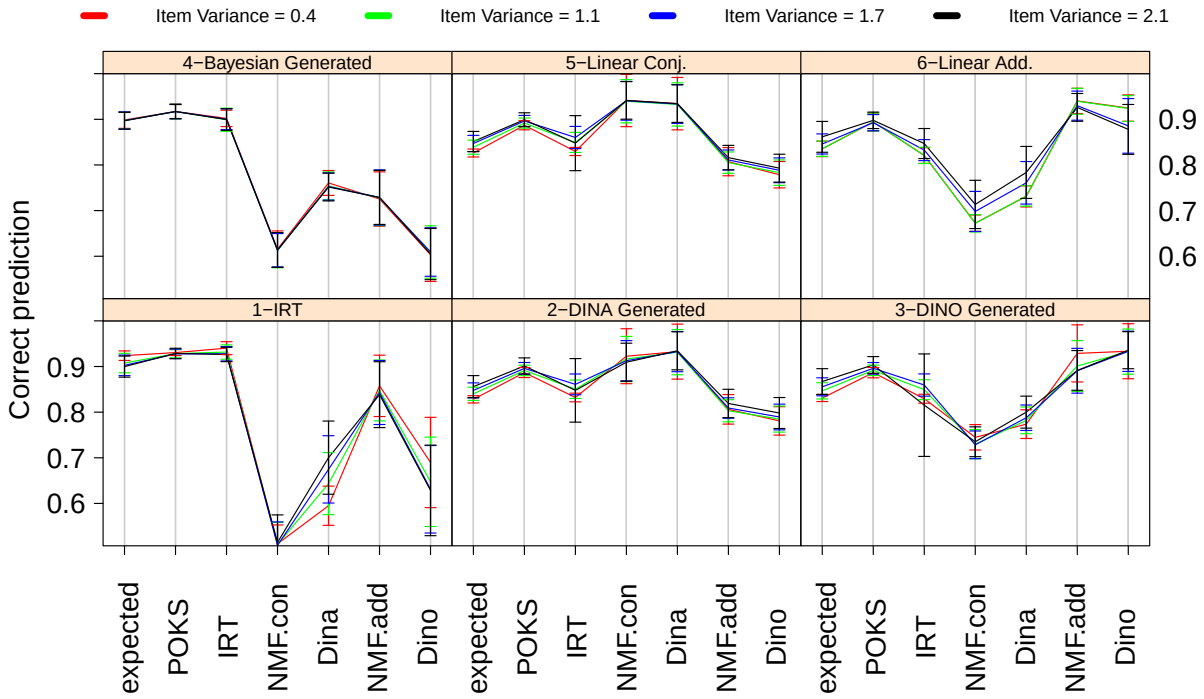


Figure 5.7 Variation of **Item Variance** Over synthetic data sets

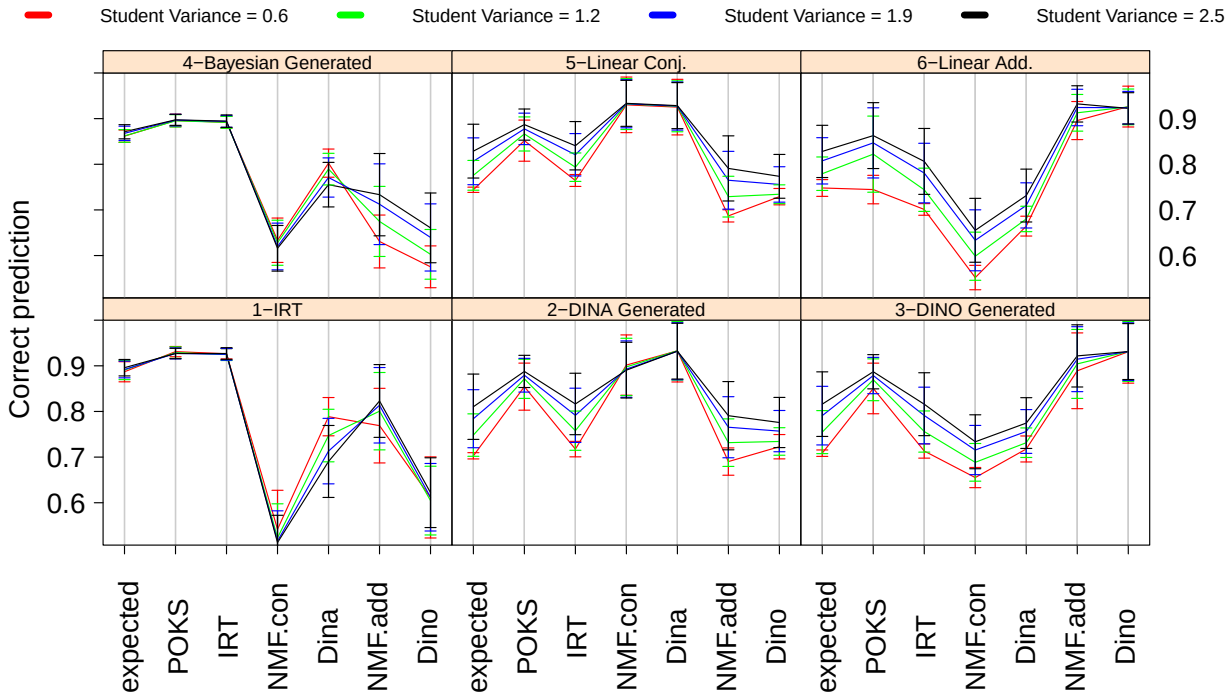


Figure 5.8 Variation of **Student variance** Over synthetic data sets

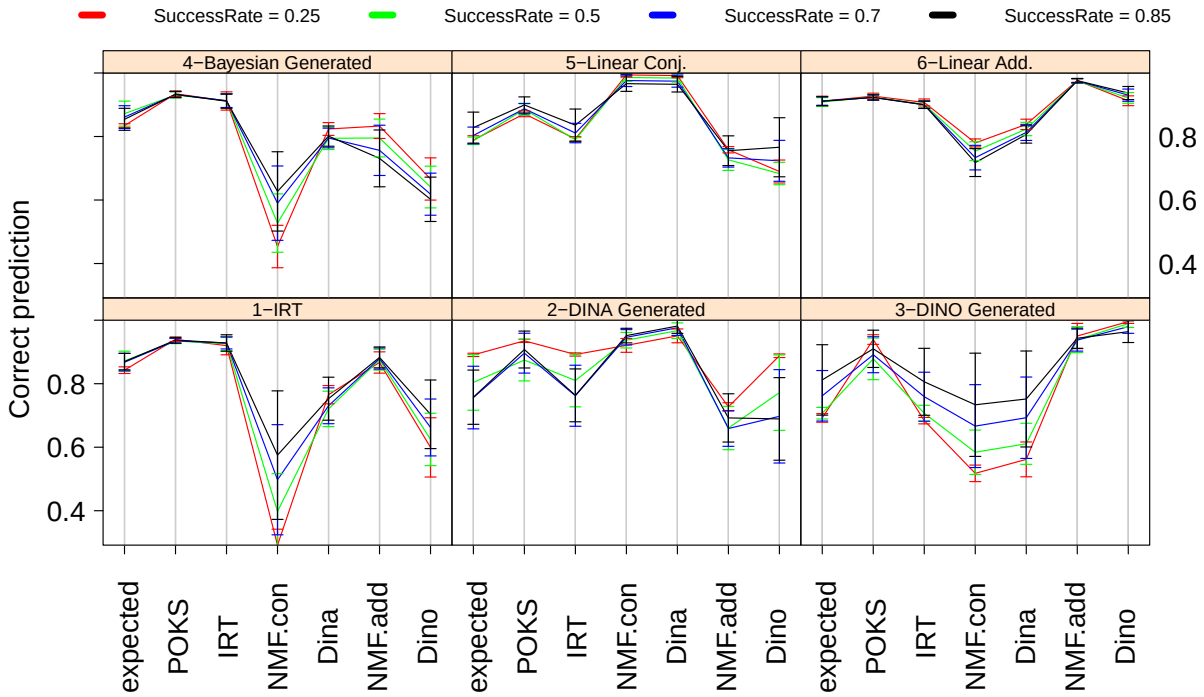


Figure 5.9 Variation of **Success Rate** Over synthetic data sets

axis as the number of skills degrades. 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. Clearly the pattern stays the same for different values except for few minor changes. Although the the set of performance values shifts across the correct prediction axis but the pattern stays the same.

Finally, we did a last experiment on datasets with different success rates. For synthetic data, the success rate starts from 0.25% up to 0.85%. The pattern is scaled for some linear methods on IRT and bayesian generated data and it doesn't change the performance of IRT and POKS themselves. For datasets with multi-skills model's ground truth the signature was shifted downward on these models predictive performance as the success rate of synthetic data degrades.

5.4 Results of Experiment 3: Model selection based on *performance vector* classification

["Prototype" is used but not explained...]

The third experiment addresses the question of identifying the ground thruth model using performance signatures. As explained before, the uniqueness characteristic of synthetic data signatures

offers a means to identify the ground truth based on the similarity between the prototype performance from a synthetic data and the target *performance vector* from the real data. In this experiment we used the measure form section 3.2.3 as a degree of similarity between these vectors.

[I started rephrasing but I do not understand very well so the rephrasing may not be correct...]

In this experiment, a number of synthetic data sets for each model are created using different parameters. There are 6 data specific parameters (namely : Number of items, students, skills, average success rate, item and student score variance) and 4 different values for each parameter. In total, this makes 24 combinations of unique parameter-value pairs. We will call a combination a **group**. For each each group, we generate 10 datasets with different seeding point for each skills assessment ground truths. In total, 24 samples are generated.

[Next paragrapy not clear... What is correlated with what? How about using mathematical notation for making this clearer? Also defining how a "prototype" is defined.]

According to the condition of the signature model selection approach we are permitted to calculate the correlation table among the members of each group. The expectation is that those vectors with same model show high correlation. Table 5.3 shows the average correlation table of these 24 groups.

5.4.1 Results

[Here too I tried to rephrase but I am not sure I have the right version and some details are missing. Please use a mathematical notation to make things clearer on how prototypes are calculated, how correlations are obtained, etc.]

We evaluate the value of the correlation between *performance vectors* as an indicator of model fit. The diagonal of table 5.3 shows the correlation between the *performance vectors* of the model that was used to create the synthetic data. As expected, it shows high correlations because it compares the same model generated datasets. On the other hand some models such as IRT and POKS also show a high correlation since they are not using multi-skills models. Those models that share concepts such as DINA and NMF conjunctive also show a high correlation comparing to 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 they are less correlated.

In general, correlation similarity provides a very good measure of model fit, although models that have strong similarities show close values that could lead to misrecognition among them, which is to be expected.

Table 5.4 shows the correlation of target *performance vector* of the real data in columns with proto-

Synthetic Datasets

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

Table 5.3 Degree of similarity between six synthetic datasets based on the correlation

Real Datasets

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

Table 5.4 Degree of similarity between six synthetic datasets and the ground truth based on the correlation

type *performance vector* 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.

5.5 Results of Experiment 4: Generality of the signature approach under different assumptions about the data

Having shown that the correlation similarity between *performance vectors* is a good indicator of model fit for synthetic data (see Table 5.3), we now move to the final experiment to evaluate the ability of the approach to identify the ground thruth model and compare the results with the simple best performer approach.

Table 5.5 shows the confusion matrix of this experiment. In total, there exists 1440 datasets where each model corresponds to 240 dataset. The gray cells in table 5.5 show the true positive values and other values in each column represent the false positive predictions for a 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 5.5 is calculated based on $\frac{TP}{240}$ which counts the true positive predictions for each sub set of datasets with the same actual ground truth. The accuracy that is reported in the last two columns of table 5.5 shows how faithful the classification method is in the sense of specificity. Therefore it counts true negative values based on $\frac{TN}{1200}$ (1200 is the number of datasets that have ground truth oppose to the target model). In terms of true positive selections there is no benefit between any of these methods even sometimes best performer shows to perform better (specially for DINA and IRT).

[Why does the last row of table5.6 has 4 grey cells?]

Taking into account the false negative and false positive values changes the classification results. Table 5.6 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 to show the comparison. The third column of each classification method shows that F-measure is increased when the signature approach is used for classification 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 5.6 shows this result. The total accuracy which considers true positive numbers over number of datasets regardless of individual models shows that the best performer approach gets 0.75% and the signature approach gets upto 0.84% of accuracy.

BP: Best Performer
NN: Nearest Neighbor

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

Table 5.5 Confusion matrix for classification of 210 synthetic datasets on 7 models with Best performer Vs. Nearest neighbor methods

Models	Performance							
	Best Performer				Nearest Neighbor			
	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 5.6 Accuracy of best performer and nearest neighbor classification methods

CHAPTER 6
Conclusion and future work

In this thesis, the performance of seven student skills assessment models is assessed and used to define a framework for evaluating mode fit. Model fit of a data set is defined as the similarity between the *performance vector* obtained over this data set and, the *prototype performance* vectors obtained over synthetic data sets.

Let us return to the conjecture that the comparison of performance vectors over synthetic data can help determine whether a specific skill model corresponds to the ground truth of some data set. As described, the standard practice is to select the model that has the highest predictive performance as the best fit. Using synthetic data, we have shown that the generative model does not always correspond to the best performer, and that an approach based on defining a vector space model of performance and on finding the nearest synthetic data point to the target data in that space provides a more reliable means to find the underlying model behind the target data.

This means of determining a data set's ground thruth is made possible because the synthetic data sets have very distinct performance patterns, showing sharp differences across models. The data sets from a single model tend to share a strong similarity among themselves, and show strong dissimilarity from most other models. This property of synthetic *performance vectors* cluster around distinct *performance prototypes* is very strong for synthetic data.

Another finding is that we find evidence that data sets that share a common source have correlated *performance vectors*. This happends with the Fraction data sets. They all have 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 a very different property, namely a single skill per item, do we observe a different *performance vector* for the models that depend on the Q-matrix (NMF conjunctive, DINA, NMF additive, and DINO).

However, the similarity of *performance vectors* does not seem to substantially extend to data that shares the same domain: although the Vomlel data is within the same domain as the Fraction data, namely arithmetic, the performance signatures are relatively different between the two. This could be attributed to the Q-matrices involved (although both are multiple skills and therefore the difference is not due to the formal property of single skill per item), but other factors could also be involved such as sensitivity to data specific parameters.

The other interesting finding is that for real data sets the performances are not better than the

expected performance and the conclusion behind that can be either the ground truth is not among the candidate models, or the best performer is not necessarily the ground truth.

The highest correlation was for Vomlel dataset which selects IRT as the ground truth. As expected , for the ECPE data set, the *performance vector* is close to that of random data which resulted in very small differences among model performances. Note that this dataset comes from English examination domain and our investigation shows that a simple query among records retrieves all possible combination of test results which is an evidence of randomness.

The results of assessing model fit for synthetic data show that some datasets with different ground truths that share some concepts, show a high correlation. For example, datasets based on linear conjunctive and DINA model. There exist no correlation for those that have completely different underlying models.

Furthermore investigations show that the predictive performance of each model over a dataset is dependent to different model and data specific parameters. For some parameters the *performance vector* stays stable but some others play an important role in changing this vector. Although the predictive *performance vector* changes for some parameters but it still shows a high correlation with datasets with the same ground truth.

The most important finding of this research is that 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. The last experiment tests the accuracy of signature approach versus best performer to classify a set of synthetic datasets with different contextual parameters. The results confirms that the "signature" approach shows a better accuracy in terms of measures for the confusion table in the span of different parameter conditions. This also shows the effect of data generation parameters on the best performer and reliability of these approaches on different assumptions about the data.

6.1 Limits

[Took out the two paragraphs because they are not limitations.

Could discuss:

1. study limited to static data as opposed to dynamic (time) data like BKT
2. also limited mostly to fraction data
3. synthetic data does not have the complexity of the real world: generality of the findings
4. dependent on the models; would it work if we only had 2 models?

]

6.2 Future Work

Further studies with real and simulated data are clearly needed. For example, there are other real datasets that potentially require a different set of candidate models. And this requires new approaches to generate synthetic data with respect to these skills assessment models.

However, the approach would generalize to dynamic data as well. This approach is using static models to assess a model fit but dynamic models are also used widely in EDM. This contribution also requires data generation and performance assessment of each model.

In this thesis we compared the "signature" approach with the "best performer" in terms of ground truth classification on synthetic data. The potential future contribution would be preparing a survey on different model selection approaches with their error metrics and comparing their performance over synthetic data. [What about likelihood or other measure of fit?]

One important future work is to give different weights to different parts of the space to justify the performance gain where in our research we assumed that the space is equally important.

Finally, because selecting right model is an important factor in all fields of studies, this approach could be applied in general fields as well. EDM is the example that we applied in our research but it can be tested in other fields of studies.

References

F. B. Baker et S.-H. Kim, *Item Response Theory, Parameter Estimation Techniques (2nd ed.)*. New York, NY: Marcel Dekker Inc., 2004.

T. Barnes, "The Q-matrix method: Mining student response data for knowledge," dans *AAAI'2005 Workshop on Educational Data Mining*, 2005, pp. technical report WS–05–02.

T. Barnes, D. Bitzer, et M. Vouk, "Experimental analysis of the q-matrix method in knowledge discovery," *Foundations of Intelligent Systems*, pp. 11–41, 2005.

T. Barnes, "Evaluation of the q-matrix method in understanding student logic proofs," dans *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006*, G. Sutcliffe et R. Goebel, éds. AAAI Press, 2006, pp. 491–496.

——, "Novel derivation and application of skill matrices: The q-matrix method," *Handbook on Educational Data Mining*, 2010.

B. Beheshti, M. C. Desmarais, et R. Naceur, "Methods to find the number of latent skills," dans *5th International conference on Educational Data Mining, EDM 2012, Chania, Greece, 19–21 June 2012*. Springer, 2012, pp. 81–86.

M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, et R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 155 – 173, 2007. [En ligne]. Disponible: http://www.sciencedirect.com/science/article/B6V8V-4MFTTBR-2/2/456f52d6d6846eb0070858cfcd5f7c40

D. M. Blei, A. Y. Ng, et M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

H. Cen, K. R. Koedinger, et B. Junker, "Automating cognitive model improvement by A* search and logistic regression," dans *Educational Data Mining: Papers from the 2005 AAAI Workshop.*, J. Beck, éd. Technical Report WS-05-02. Menlo Park, California: AAAI Press, 2005, pp. 47–53.

——, "Learning factors analysis — A general method for cognitive model evaluation and improvement," dans *Intelligent Tutoring Systems, 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006, Proceedings*, 2006, pp. 164–175.

C.-Y. Chiu, "Statistical refinement of the Q-matrix in cognitive diagnosis," *Applied Psychological Measurement*, 2013.

A. T. Corbett et J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1995.

C. Cortes et V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

J. De La Torre, “An empirically based method of Q-Matrix validation for the DINA model: Development and applications,” *Journal of educational measurement*, vol. 45, no. 4, pp. 343–362, 2008.

M. Desmarais, “Conditions for effectively deriving a Q-matrix from data with non-negative matrix factorization,” dans *4th International Conference on Educational Data Mining, EDM*, 2011, pp. 41–50.

M. Desmarais, B. Beheshti, et P. Xu, “The refinement of a q-matrix: Assessing methods to validate tasks to skills mapping,” dans *Educational Data Mining 2014*, 2014.

M. C. Desmarais, “Mapping question items to skills with non-negative matrix factorization,” *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 2, pp. 30–36, 2012.

M. C. Desmarais et R. Baker, “A review of recent advances in learner and skill modeling in intelligent learning environments,” *User Modeling and User-Adapted Interactions*, vol. 22, pp. 9–38, 2011.

M. C. Desmarais et R. S. d Baker, “A review of recent advances in learner and skill modeling in intelligent learning environments,” *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 9–38, 2012.

M. C. Desmarais et R. Naceur, “A matrix factorization method for mapping items to skills and for enhancing expert-based Q-Matrices,” dans *6th International Conference, AIED 2013, Memphis, TN, USA*, 2013, pp. 441–450.

M. C. Desmarais et I. Pelczer, “On the faithfulness of simulated student performance data,” dans *3rd International Conference on Educational Data Mining EDM2010*, R. S. J. de Baker, A. Merceron, et P. I. Pavlik, éds., Pittsburgh, PA, USA, June 11–13 2010, pp. 21–30.

M. C. Desmarais et X. Pu, “A bayesian inference adaptive testing framework and its comparison with Item Response Theory,” *International Journal of Artificial Intelligence in Education*, vol. 15, pp. 291–323, 2005.

M. C. Desmarais, A. Maluf, et J. Liu, “User-expertise modeling with empirically derived probabilistic implication networks,” *User Modeling and User-Adapted Interaction*, vol. 5, no. 3-4, pp. 283–315, 1996.

——, “User-expertise modeling with empirically derived probabilistic implication networks,” *User Modeling and User-Adapted Interaction*, vol. 5, no. 3-4, pp. 283–315, 1996.

M. C. Desmarais, P. Meshkinfam, et M. Gagnon, “Learned student models with item to item knowledge structures,” *User Modeling and User-Adapted Interaction*, vol. 16, no. 5, pp. 403–434, 2006.

A. Dhanani, S. Y. Lee, P. Phothilimthana, et Z. Pardos, “A comparison of error metrics for learning model parameters in bayesian knowledge tracing,” Technical Report UCB/EECS-2014-131, EECS Department, University of California, Berkeley, Rapp. tech., 2014.

J.-P. Doignon et J.-C. Falmagne, “Spaces for the assessment of knowledge,” *International Journal of Man-Machine Studies*, vol. 23, pp. 175–196, 1985.

——, *Knowledge Spaces*. Berlin: Springer-Verlag, 1999.

P. Domingos et M. Pazzani, “On the optimality of the simple Bayesian classifier under zero-one loss,” *Machine Learning*, vol. 29, pp. 103–130, 1997.

M. Feng, N. Heffernan, C. Heffernan, et M. Mani, “Using mixed-effects modeling to analyze different grain-sized skill models in an intelligent tutoring system,” *Learning Technologies, IEEE Transactions on*, vol. 2, no. 2, pp. 79–92, 2009.

I. Guyon et A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

T. Hastie, R. Tibshirani, J. Friedman, et J. Franklin, “The elements of statistical learning: data mining, inference and prediction,” *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

B. W. Junker et K. Sijtsma, “Cognitive assessment models with few assumptions, and connections with nonparametric item response theory,” *Applied Psychological Measurement*, vol. 25, no. 3, pp. 258–272, 2001.

K. R. Koedinger, A. T. Corbett, et C. Perfetti, “The knowledge-learning-instruction (KLI) framework: Toward bridging the science-practice chasm to enhance robust student learning,” CMU-HCII Tech Repor, Carnegie-Mellon University, Human Computer Interaction Institute, Rapp. tech., 2011.

Y. Koren, R. Bell, et C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

D. Lee, H. Seung *et al.*, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

R. P. Lippmann, “An introduction to computing with neural nets,” *ASSP Magazine, IEEE*, vol. 4, no. 2, pp. 4–22, 1987.

N. Loye, F. Caron, J. Pineault, M. Tessier-Baillargeaon, C. Burney-Vincent, et M. Gagnon, “La validité du diagnostic issu d’un mariage entre didactique et mesure sur un test existant,” *Des mécanismes pour assurer la validité de l’interprétation de la mesure en éducation*, vol. 2, pp. 11–30, 2011.

A. Robitzsch, T. Kiefer, A. George, A. Uenlue, et M. Robitzsch, “Package CDM,” 2012, http://cran.r-project.org/web/packages/CDM/index.html.

R. B. Rosenberg-Kima et Z. A. Pardos, “Is this data for real?” dans *twenty years of knowledge tracing workshop*, 2014, pp. 141–145.

——, “Is this model for real? simulating data to reveal the proximity of a model to reality,” dans *2nd AIED Workshop on Simulated Learners*, 2015.

D. Seung et L. Lee, “Algorithms for non-negative matrix factorization,” *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

R. Sison et M. Shimura, “Student modeling and machine learning,” *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 9, pp. 128–158, 1998.

K. K. Tatsuoka, “Rule space: An approach for dealing with misconceptions based on item response theory,” *Journal of Educational Measurement*, vol. 20, pp. 345–354, 1983.

K. Tatsuoka, *Cognitive Assessment: An Introduction to the Rule Space Method*. Routledge Academic, 2009.

K. Tatsuoka, U. of Illinois at Urbana-Champaign. Computer-based Education Research Laboratory, et N. I. of Education (US), *Analysis of errors in fraction addition and subtraction problems*. Computer-based Education Research Laboratory, University of Illinois, 1984.

N. Thai-Nghe, T. Horváth, et L. Schmidt-Thieme, “Factorization models for forecasting student performance,” dans *Proceedings of EDM 2011, The 4th International Conference on Educational Data Mining*, C. Conati, S. Ventura, M. Pechenizkiy, et T. Calders, éds. www.educationaldatamining.org, Eindhoven, Netherlands, July 6–8 2011, pp. 11–20.

H.-T. Trinh, “Educational data synthesizer,” 2015, p. technical report at Ecole Polytechnique de Montreal. [En ligne]. Disponible: https://github.com/thtrieu/edmsyn/tree/master/vignettes

J. Vomlel, “Bayesian networks in educational testing,” *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, vol. 12, pp. 83–100, 2004.

T. Winters, “Educational data mining: Collection and analysis of score matrices for outcomes-based assessment,” Thèse de doctorat, University of California Riverside, 2006.