Modeling Techniques for Adaptive Practice Systems

Doctoral Thesis

Jiří Řihák

Brno, Fall 2017
Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Jiří Řihák

Advisor: doc. Mgr. Radek Pelánek Ph.D.
Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor doc. Radek Pelánek for the continuous support, motivation, enthusiasm, and immense knowledge. His guidance helped me uncover mysteries of machine learning and data analysis. I could not have imagined having a better advisor and mentor for my Ph.D. study. I would also like to thank my colleague Jan Papoušek who was always there ready to discuss our research, to inspire me and to help me hunt bugs. Last but not least, I would like to thank my family and my life partner for their unconditional support during my studies.
Abstract

We study the challenges associated with the development of adaptive practice systems, particularly with the focus on the description of methods, which allow running such intelligent systems which give answers to questions occurring during development and automatize tasks usually done by human experts.

First, we describe various techniques which help to create a domain model that describes the knowledge domain on which the system focuses. Some of these techniques allow building a domain model automatically from the learners’ data, some of them combine the data-driven approach with input from a human expert and some process data and give experts an insight into the domain and foundations for further decisions. Next, we depict several learner models which allow the system to adapt to every user and provide individualized content. Finally, we deal with response times as extra information to the frequently used correctness of answers. We study how to combine these two types of information and how the addition of the response time benefits the models and the analysis.

To evaluate these techniques, we use both simulated data and datasets from various real-world systems. To study response times, we developed the MatMat system focused on the basics of mathematics; that is the domain where response time plays a significant role in measuring the learner’s success. The reported results provide valuable insight for both developers of adaptive practice systems and researchers interested in learner and domain modeling.
Keywords

intelligent tutoring system, adaptive practice, system development, domain modeling, learner modeling, response times, knowledge components, clustering, visualization, similarity metrics, elo rating system
Contents

1 Introduction ...................................................... 1
   1.1 Intelligent Tutoring Systems .............................. 1
   1.2 Development of Intelligent Tutoring Systems ........... 2
   1.3 Case Study - MatMat ....................................... 3
   1.4 Contribution of the Thesis ............................... 4

2 Systems for Adaptive Practice ............................. 7
   2.1 Adaptive Practice ......................................... 7
   2.2 Adaptive Learning Group’s Systems ....................... 9
      2.2.1 Outline Maps ......................................... 10
      2.2.2 Anatom ................................................. 12
      2.2.3 Umíme Česky .......................................... 13
      2.2.4 Problem Solving Tutor ............................... 13
   2.3 MatMat ....................................................... 14
      2.3.1 Main Idea .............................................. 14
      2.3.2 Main Components ..................................... 16
      2.3.3 Implementation Notes .............................. 20
   2.4 Related Systems .......................................... 21
   2.5 Datasets ..................................................... 23
      2.5.1 Real-World Data .................................... 23
      2.5.2 Basic Statistics ...................................... 26
      2.5.3 Simulated Data ..................................... 27

3 Domain Modeling .................................................. 29
   3.1 Related Work .............................................. 30
      3.1.1 Q-matrix ............................................. 30
      3.1.2 Clustering Methods .................................. 32
      3.1.3 Dimension Reduction Techniques .................... 34
   3.2 Similarity Measures ....................................... 37
      3.2.1 Similarity Measures Overview .................... 38
      3.2.2 Second Level of Similarity Measure .............. 40
      3.2.3 Comparison of Similarity Measures .............. 41
      3.2.4 Do We Have Enough Data? ......................... 48
      3.2.5 Similarity Measures with Solving Times .......... 48
   3.3 Automatic Concept Detection ........................... 51

vii
1 Introduction

1.1 Intelligent Tutoring Systems

Nowadays information technologies are changing almost all aspects of human life including education. Education is historically a domain of teachers, but the connection with IT brings new methods which can move the whole process of learning forward. Computer systems that apply these methods are called Intelligent tutoring systems (ITS) [119, 85]. These systems utilize the ability of computers to collect and process large amounts of data about learner interaction with the system and combine them with knowledge about an educational area.

In some instances (e.g., Duolingo\textsuperscript{1}) the system has the ambition to be the main element in the educational process, but ITS are mostly designed to be a complement to traditional education with a human tutor. Good ITS can be an excellent tool for teachers because it can save the teacher’s time on routine tasks such as evaluation of homework, or give detailed information of the learner’s skill, struggles, progress, etc. This kind of help allows teachers to focus more on tasks which require human reasoning and empathy instead of repetitive tasks.

An ITS can take distinct forms and different strategies to produce some type of intelligent behavior. In this work, we focus mainly on the systems where the main aim is an adaptation to a learner. The main idea is to individualize the presented content to every learner, so the learning process is more efficient. One of the examples of this behavior is the adjustment of difficulty of the presented tasks in such a way that learners do not lose time with tasks that are too easy or beyond their skills.

We call this type of educational strategy adaptive learning. The main advantages of the adaptive approach are a reduction of time required to master the area and higher learners’ motivation due to an appropriate choice of content. Moreover, adaptive learning can address the common educational issue of significant skill differences in class. Teachers often do not have time to guide slower students or

\begin{footnotesize}
\begin{enumerate}
\item \url{https://www.duolingo.com/}
\end{enumerate}
\end{footnotesize}
prepare challenging tasks for the faster ones and an adaptive learning system can partially compensate for this limitation.

1.2 Development of Intelligent Tutoring Systems

Developing such a system brings many challenges from both the theoretical and practical point of view. Developers have to choose the program language, platform or framework, design the database structure and user interface and answer many other practical questions. However, in this thesis, we focus mainly on theoretical questions, such as: What data to collect and how to use them? How to model a learner’s knowledge - which skills do we use? How is the educational content of the system related to skills? How to adaptively select content for a learner? How to properly set meta-parameters of models and algorithms in the system? Even though our focus is mainly on theoretical questions, we do all our reasoning and conclusions with practical usage in mind. A learner model with excellent results which is not possible to use online in a system is not practical.

Machine learning [12] and current research offers vast numbers of methods, models and tools for learning analysis from which a developer or a researcher of an adaptive system can choose. But the question remains, which methods to implement into the system and which ones to use in learning analysis? And even after the selection, there are many questions to answer. Different techniques typically have many variants or parameters to set, and an appropriate choice of these is often more important than the choice of the method.

There are no general answers to these questions because every context is specific. Every adaptive learning system has different goals, type of learners, content, interface, desired adaptive behavior, etc. Every context needs a unique approach and set of methods that reflect its specifics. Therefore it is important to search for guidelines and evaluation methods which allow developers of an adaptive system to make the right choices. This thesis describes several classic and novel methods but also searches for general approaches, advice, and conclusions which are usable by other developers and researchers dealing with a similar type of challenges.
1.3 Case Study - MatMat

This work is partially based on experience in developing our own adaptive practice system called MatMat. This system is focused on building fluency in basic mathematical operations (counting, addition, multiplication, etc.) by presenting adaptively selected simple tasks in a rapid manner. The idea was to create a system with a simple interface but advanced adaptive behavior. Our approach was as much data-driven as possible due to two reasons. Firstly, a lack of resources to get high-quality domain expert input, and secondly, the desire to test the limits and advantages of machine learning methods on these tasks.

The domain for this system was chosen not only for common problems of children and adults with mathematics but also for the specificity of this domain. Thanks to the focus on building fluency, response time has an important role in estimating the abilities of learners. The topic of response time utilization in educational data mining does not receive enough attention and the data from the MatMat system provide a good opportunity to study new techniques and methods of comparison.

The three main areas included in this thesis are automatic domain modeling, learner modeling, and utilization of response times. These three areas arise directly from the needs of developing such a system like MatMat. Domain modeling is needed to organize educational content of the system, and it is an input to advanced learner modeling methods. Learner modeling is a key method for the adaptive behavior of the system and allows to capture the learner’s abilities. Finally, utilization of the response time is specific for this type of system, which seems to be especially important due to the motivation of building fluency in basic arithmetic tasks. In such a case, the correctness of answer – which is usually used in a system with simple tasks – does not seem sufficient information to properly capture the learner’s abilities.
1. Introduction

1.4 Contribution of the Thesis

Here we give an overview of the main contributions of this thesis with references to the parts of the thesis and our publications which cover the given topic.

Domain Modeling

We propose a two step approach of domain modeling: computation of similarity of items and follow-up techniques with a different focus such as automatic concept detection or visualization. The main contributions in this area are as follows.

- Overview of item similarity measures and evaluation of their suitability for learning analysis with specific recommendations (Section 3.2, [111]).

- Introduction of the second level of the similarity measure, which significantly improves the quality of computed item similarities (Section 3.2, [111]).

- Introduction of the method which allows us to combine the data-driven approach of building a domain model and the domain model provided by a human expert. This approach based on supervised learning methods allows us to detect mistakes in experts’ labeling. We also discuss conditions under which this approach is most beneficial (Section 3.4, [83, 82]).

- Overview of standard and recent clustering and visualization approaches and the evaluation of their applicability in the educational data mining. (Sections 3.3 and 3.5, [16]).

Learner Modeling

In this area, we focus mainly on Elo based models. The main contributions in this field are as follows.

- Introduction of a concept and a network model for prior knowledge estimation, which allows utilizing the domain model (the concept model) or item similarities (the network model). These
models perform better than the previously used basic Elo model and give more detailed information about learners’ skills (Section 4.2, [84, 95]).

- Introduction of the hierarchical model, which deals with both prior and current knowledge of learners. This model works with the more complex (hierarchical) domain model and allows us to track very specific skills as well as estimate and update these skills quickly. This model performs better than the previously used PFAE model, and it was deployed on the MatMat system (Section 4.3, [108, 110, 109]).

Response Times

We explored the usability of response times as an additional source of information for various ends. We showed that the response time carries interesting information about learners and their skills (Section 5.2, [86]). We introduced various ways to combine the response time and correctness of answers to a single measure of success (Section 5.3, [108, 110, 109]). This approach has the following benefits.

- The new combined measure of success redefines success in such a way that the speed of learners has a significant role. This approach is especially crucial for systems for building fluency in certain tasks.

- The use of response time gives a different point of view on the similarity of items in some domains or improves the stability of similarities, i.e., decreases the size of the dataset needed to get good item similarities (Sections 5.4 and 5.5, [97]).

- The response time provides an alternative point of view on mastery detection (Section 5.5.2, [111]).

Evaluation Methods

In this work, we also focused on the use of diverse methods to evaluate the presented techniques. The main contributions in this area are as follows.
1. **Introduction**

- Overview of evaluation methods with their advantages and limitations (Section 6.2.2).

- We address issues connected to the comparison of learner models with and without the use of response time (Section 5.4.1, [109]).

- Introduction of an evaluation method focused on the quality of parameters and stability of results which gives an alternative comparison of techniques (Sections 3.2.3 and 5.4.2, [111, 109]).

- Introduction of an evaluation method which allows us to estimate sufficiency of the size of the used dataset for a given analysis (Sections 3.2.4 and 5.5, [111]).

**MatMat**

For an analysis of domain and learner models and especially of response times, we developed, and described in this work, the system called MatMat (Section 2.3). It is the first adaptive system to practice basic mathematical operations in the Czech language. This system was the source of data not only for this work but also for other bachelor and master theses and research papers. Moreover, it allowed us to use some of the techniques described in this thesis in practice (e.g. hierarchical model or response time utilization), which led to an improvement of the adaptive behavior of the system.
2 Systems for Adaptive Practice

2.1 Adaptive Practice

There is a whole spectrum of intelligent tutoring systems which vary in goals, content, functionality, type of intelligent behavior, target audience and other aspects. As mentioned in the introduction, we work with adaptive learning systems where the priority is system individualization to learners. Moreover, we focus on adaptive practice. In such systems, the goal is not to explain a part of the curriculum in its complexity but practice already known skills or help to memorize facts that are necessary to be learned by heart. A good example of such a topic is the vocabulary of a foreign language.

Terminology

Figure 2.1 shows the essential terms used in this work:

Learner is a user who interacts with a system regardless of whether he or she is a child, a student or an adult.

Item is a task practicing a part of the educational content. In our setting, items are mostly atomic and indivisible, and we are interested in their difficulty. Examples of such tasks are

- translation of an English word to German,
- correct spelling of a word,
- the name of a city and its location on the map,
- $6 \times 7 = ?$.

Concept (also known as knowledge component) is a part of knowledge or an ability. In a system, a concept is formed by a set of items. A concept can be very general (algebra, geometry, European cities) or very specific (fraction simplification, multiplying 2 and 3). A skill is then the degree of the learner’s mastery of a specific concept.
2. Systems for Adaptive Practice

**Question** is a presentation of an item. One item can be presented in multiple ways, e.g., as open-ended or as a multiple-choice question with various options.

**Answer** is a response of a learner to a question. We are mostly interested in

- **correctness** — binary information about the learner’s success (sometimes simply called response),
- **response time** — time needed by a learner to answer the question.

![Diagram of user interaction with a system during practice.](image)

Figure 2.1: Simplified schema of user interaction with a system during practice.

**Components of Adaptive Systems**

Intelligent tutoring systems usually consist of four basic components [121]. The first one is the **domain model**, which defines concepts, knowledge, and strategies of the topic being learned. Chapter 3 of this work is devoted to this topic, especially how to analyze and build such a domain model. The next component is the **learner model**, which consists of the cognitive, motivational and other psychological states that are estimated from solving data during interaction with the system. In this work (Chapter 4), we mainly focus on modeling learners’ skill rather than their motivational state.

The third component is called the **tutoring model** and takes the domain and learner models as an input and selects what the system should do next to move the learner state to more optimal states in
2. Systems for Adaptive Practice

In adaptive practice, this is typically a selection of an appropriate item for practice. The last component is the user interface, which works with the other three components and provides interaction between the learner and the system.

The possible actions of the system suggested by the tutoring model can be divided into two loops: the outer and the inner loop [131]. The outer loop is focused on the divisions of items into units, the selection of suitable problems for learners and the order in which they are presented. The inner loop is responsible for providing the hints and feedback within a problem in a way that the learner can complete the task. With regard to the atomicity of the items in adaptive practice, we focus primarily on the outer loop.

The main goal of an adaptive practice system within the outer loop is the selection of suitable educational content in an appropriate form. The system should present challenging and interesting items for more skilled learners and, on the other hand, easier items, or guidance for the less skilled. This intelligent selection of content is an important aspect of the maximization of the learning effect and motivation of learners. An appropriate selection of items keeps a learner in the flow state [25, 26]) where a learner is neither bored nor overwhelmed.

Another important part of adaptive practice is to provide informative feedback for learners or teachers. Good feedback and information about progress is an important part of motivation and serves as guidance which topic should be practiced next. Every learner should master each unit before proceeding to more advanced topics independently of his or her speed. Therefore, a good understanding of learners’ skills is necessary. This concept of learning is called mastery learning [13]. This approach is often in contradiction with traditional educational practice where the time for a topic is limited. Adaptive practice systems should provide an individual length of practice and recognize the right moment to move to the next topic.

2.2 Adaptive Learning Group’s Systems

The author of this thesis is a member of the Adaptive Learning Group1, a small research lab at the Faculty of Informatics, Masaryk University.


9
2. Systems for Adaptive Practice

One of the activities of this group is the development of freely available adaptive practice systems which serve as a tool for testing new techniques, randomized controlled group trials and, especially, as a source of educational data for the learning analysis.

These systems are also the main sources of data for this work. For a better understanding of the used datasets, we give a short description of these systems. The next section is dedicated to a more detailed description of the MatMat system, which is the most important system for analysis in this thesis.

All systems created by the Adaptive Learning Group focus mainly on the outer loop of practice. The systems work with simple atomic tasks which are presented in rapid succession. The goal of this type of practice is memorization of the educational content or getting fluent in simple tasks, often a combination of these two goals. First, a learner learns a skill or memorizes a fact and then he or she gets fluent in the application of the skill or in the retrieving of the fact.

The type and the use of adaptability of the systems can differ, but mostly they are focused on the selection of tasks presented to the learner and feedback in the form of a visualization of the learner’s skill. The main idea, however, is always the same - to implement techniques that can work online in a system and improve the process of learning and user experience.

The general schema of these systems is shown in the diagram in Figure 2.2, which depicts the main components of the systems. The main practice loop is formed by the learner model which provides the probabilities of a correct answer to the available items. This information is used by the item selection component which selects the item presented to the learner. After the learner answers, his or her response is processed by the learner model that updates its estimations and closes the loop.

2.2.1 Outline Maps

Outline maps is a system² focused on the practice of geography knowledge. In this system, learners practice names of geographic places such as states, rivers, cities, seas, mountains, islands, etc. The systems

². https://outlinemaps.org
offers a wide variety of practice contexts such as European states, European cities, African states, Czech rivers, Czech districts and more than 50 others. A practice session consists of sets of 10 questions from a context selected by a learner. Every question works with an interactive map, and there are two possible types: What is the highlighted place? or Choose given place on the map. The first type of question is always multiple choice and learners have to choose the correct answer from 2 to 6 options. The second type of question can be either a multiple choice or an open question for which the learner can choose from all the places on the map.

The adaptability [89] of this systems mainly lies in adjusting the difficulty of tasks which are presented to a learner. This is achieved by two techniques. The first one is the selection of an item with appropriate difficulty. The second technique is a customization of the number of options in multiple choice questions to adjust the difficulty even more. The system aims at a target difficulty of a 65% success rate (75% in an older version of the system) to make practice challenging but not frustrating.
2. Systems for Adaptive Practice

The system is available in six languages including English and Czech but the majority of learners (95%) come from Czechia and Slovakia, countries with similar languages and background. The system is also used in schools as part of geography education. There is no explicit information whether a learner is using the system as part of a class or in their free time, but it is partially possible to detect the use of the system in a class from IP addresses and time of access.

Authentication is possible but not required, and most of the learners do not sign up to the system. The system uses HTTP cookies to remember the returning learners to maximize individualization of the practice, but this method has its limits and many learners are recorded in the data under multiple identities.

2.2.2 Anatom

Anatom is a system\(^3\) with a very similar functionality to Outline maps but with a different topic. Anatom is focused on the practice of medical anatomy. The maps are replaced by pictures of organ systems and body parts, but the task remains the same: match the name of a term (or place) with the corresponding part of the picture (or map). The target audience of this system is also specific: students of medicine. This narrower focus limits the usage of the system but learners are typically more motivated to use the system repetitively.

3. [https://practiceanatomy.com](https://practiceanatomy.com)
2. Systems for Adaptive Practice

The system is available in Czech and English, but the most practiced terms are in Latin. The system also offers other types of practice such as the practice of term relations, however, this work uses only the data from the basic practice.

2.2.3 Umíme Česky

The Umíme Česky\(^4\) system focuses on the practice of Czech grammar rules, which is a relevant topic from elementary to high school and some topics can also be useful for adult learners. The system contains several different types of exercises and educational games. In our analysis, we used the data from basic fill-in exercises where learners have to choose from two options to fill in a missing letter or letters in a word or a phrase. An example of such an exercise is choosing between \(i\) or \(y\) which are frequently confused letters or choosing between small or capital letters.

This system uses a different type of adaptability than the systems described above or MatMat. The practiced items are organized by grammar topic and up to three difficulty levels. After a learner chooses a topic, items are presented in a random order, but the length of the practice session is controlled. The system uses the correctness of the answer to measure the skill of the learner and gives immediate feedback about the level of mastery.

The system is used in schools and offers support for teachers and homework (e.g., reach mastery in selected topics). Given the nature of the subject, the system is used only by Czech learners.

2.2.4 Problem Solving Tutor

The Problem Solving Tutor\(^5\) is older than other systems and has quite a different type of content. The system does not use simple items but more complex tasks, which can take minutes or dozens of minutes to solve. The main difference is that the measure of a learner’s success is not the correctness of answers (learners have an unlimited number of attempts) but the time needed to finish the task. Originally, the system started with logic puzzles, e.g., Sokoban or Rush hour, for

\(^4\) http://www.umimecesky.cz
\(^5\) https://tutor.fi.muni.cz
which solving time is a natural demonstration of skill. Later, some educational problems were added, but the focus remained on high school and adult learners.

The system implements basic adaptability. Feedback about performance is presented to the learner after finishing the task, and it is based on the estimated skill of the learner. The system additionally offers two tasks of the same type which should keep learners in the flow state.

The system is a source of interesting data for the study of solving time, which can be applied to other systems where the correctness of answers is the main source of information but response time can be used as a secondary source of information.

2.3 MatMat

This system\textsuperscript{6} has a similar approach to practice and a similar functionality to other systems, especially Outline maps, but the domain of basic arithmetic has specifics which deserve more attention. Moreover, data from this system are essential to the study of response time which is an important part of this thesis.

2.3.1 Main Idea

The main goal of MatMat is to lead learners to fluency in arithmetic tasks such as counting and the basics of addition, subtraction, multiplication and division. In this case, basics means only integers less than 100, small multiplication table, etc. Multiple studies suggest \cite{112, 100, 48} that fluency in these basic tasks is essential to the mastery of more advanced mathematical topics, which requires full mental capacity.

Therefore, the task is not only to lead the learner from the \textit{unknowing state} to the \textit{knewing state} which is typical for the practice of facts such as geographical places. As the diagram in Figure 2.5 shows, the goal is also to move the learner form the \textit{knewing state}, when the learner understands the process necessary to finish the task successfully, to

\begin{itemize}
\item \textsuperscript{6} \url{https://matmat.cz}
\end{itemize}
2. Systems for Adaptive Practice

Figure 2.4: Practice of multiplication with MatMat.

the fluent state where the learner can finish the task without thinking hard.

For example, consider the task \(3 + 5\). At first, children in the not knowing state do not know how to solve this task, but then they learn how addition works and can calculate the answer 8 using their fingers and move to the knowing state. But this is not enough. They are faster and faster with more practice, and after a while, they learn the answer by heart and finally move to the fully fluent state.

To achieve fluency, the system works only with basic atomic tasks which the system offers in rapid succession. The importance of fast answers is reflected in the user interface, fast answers are emphasized in gamification and have an impact on learners’ estimated skills. The
whole learning process is supported by multiple graphical representations of tasks (e.g., number-line, counting objects, etc.), which is important in building number sense [2, 118].

2.3.2 Main Components

In this section, we describe the main components of the system shown in Figure 2.2.

Domain Model

The goal of the domain model in the system is to organize a large number of items in a structure and describe relations between them. The domain model is used in the system in two main ways. In the first one, the relations between items serve as input to the learner model and allow for more precise predictions. In the second one, the domain model organizes items to concepts and sub-contexts which are used in the user interface and help create meaningful and comprehensible feedback.

Consider the following example. A learner understands the principles of multiplication of one digit numbers. He or she is fully fluent in simpler tasks such as $2 \times 3$ or $5 \times 6$ but some harder tasks like $7 \times 6$ or $7 \times 8$ are still challenging. To allow the learner model to capture this situation in sufficient detail and select suitable items to practice, the domain model needs to distinguish small concepts corresponding to individual tasks such as $7 \times 6$ or $7 \times 8$. On the other hand, these tasks are not completely independent and the domain model needs to capture this relation.

The used model is shown in Figure 2.6. The concepts are organized in a tree-like structure with the general math skill as the root and the children of each concept are its sub-concepts. The concepts in the first level are arithmetic operations. The second level consists of sub-concepts which typically differ in the size of the used integers. The most detailed concepts which correspond to individual arithmetic tasks are placed in the last level. Note that not all level 2 concepts are further divided into sub-concepts. This approach was used only for the simplest tasks where we expect full fluency, e.g., multiplication of one digit numbers or addition up to 20. More complicated tasks such
2. Systems for Adaptive Practice

Figure 2.6: Hierarchical model used in MatMat.

as multiplication with numbers larger than 10 were grouped to one concept without sub-concepts.

The items are then assigned to one of the level 2 or 3 sub-concepts. In this way, every sub-concept consists of several items. In case of the level 3 sub-concepts, there are up to 10 different items which are similar expressions (e.g., $2 + 3$ and $3 + 2$) or different graphical representations of the same task. In the case of the level 3 sub-concepts, there can be dozens or hundreds of items.

This structure makes it possible to describe the learner’s knowledge in great detail where necessary and, through the tree structure, propagate information to near concepts. This allows the learner model to estimate a learner’s skill even after a small number of answers in the system.

Learner Model

The goal of this component is to estimate a learner’s skill for each of the concepts from the domain model and difficulties of all items in the system. Its main functionality is calculating the probability of a correct answer for a given item and learner. This information is then used in item selection or can be aggregated over the concepts and used as a source of feedback for the learner or teacher.

The learner model is designed to work with only a minimum of prior information which would have to be delivered by experts
2. Systems for Adaptive Practice

or developers. External input to the learner model consists of the domain structure provided by the domain model and some meta parameters set by developers. All other necessary parameters (e.g., item difficulties) are estimated automatically by the model and online from data learner interactions. A more detailed description of the used learner model can be found in Section 4.3 about the hierarchical model.

Item Selection

The learner chooses a concept (multiplication) or a sub-concept (small multiplication table) which he or she wants to practice. The function of the item selection component is to select from the pool of all available items in the concept those which will be presented to the learner. This component uses a similar approach to the one used in the Outline maps system which was described in [88].

The selection of an item has to balance three aspects:

- We want to keep the learner in the flow state; thus item selection prefers items with the probability of a correct answer closer to the selected target probability (target probability is set to 0.7).

- We want to make the practice diverse; thus item selection prefers never practiced or rarely practiced items over often practiced ones.

- We want to avoid repeating the same task in a short time; thus items which were answered recently are penalized. Also, items which are similar, e.g., different graphical representation of the same task, are also penalized.

All these aspects are a basis for 3 item scores. The sum of these partial scores is the total score, and the item with the maximal total score is presented to the learner. All three aspects are changing with new answers thus it is necessary to recompute the scores after every answer.
2. Systems for Adaptive Practice

Feedback

MatMat gives two types of feedback to learners: immediate one and an overview of the learner’s skills. Immediate feedback is given after every answer and the system displays the correctness of the learner’s response and the correct answer if the learner’s answer was wrong. Additionally, the learner obtains praise if the answer was correct and fast.

An example of an overview of a learner’s skill is shown in Figure 2.7. The learner can get information about his or her skill in concepts in all levels. The estimated skill is visualized by the color transition from red meaning low skill to green meaning mastery. As it can be seen in the figure, some parts of the small multiplication table are still transparent, these parts have not been practiced yet. However, the system has already made the same estimation based on other answers, and weak coloring is shown.

![Figure 2.7: Example of feedback on the small multiplication table.](image)

19
2. Systems for Adaptive Practice

This feedback is shown after every practice set, which consists of 10 items, so the learner can watch his or her progress and select next concept to practice.

Teacher Support

The system offers basic support for teachers. Learners’ accounts can be linked to the teacher’s account and the teacher can organize them into classes. To facilitate the use of the system in a class with small children who are not able to handle their own account, MatMat provides the option to create and authenticate a learner’s account entirely using the teacher’s account. Later, the child’s account can be shared with parents to allow practice at home.

Figure 2.8 shows a simple overview of learners in class in the teacher mode. The teacher can see here, what his or her learners practiced and what their strengths and weaknesses are.

![Figure 2.8: Skill overview in teacher mode.](image)

2.3.3 Implementation Notes

The whole project is open sourced, and is available at Github. The system is based on a common core called proso-apps which is used by other systems (e.g., Outline maps and Practice anatomy).

The back-end is built on Django, a Python web framework, and only provides REST API used by the front-end. The front-end is built

with Angular, a Javascript framework, with the use of HTML5 and CSS3 standards.

2.4 Related Systems

Here we give an overview of the main systems with a focus analogous to MatMat, which were an inspiration in the development of similar systems for the Czech environment.

ASSISTments

ASSISTments [43, 53, 103, 127] is an online tutoring platform that thoroughly works with the inner loop (instructional assistance, immediate feedback, etc.) and offers intelligent exercises from mathematics and other domains. ASSISTments is free to use in schools and supports cooperation of teachers and learners in the learning process (e.g., assigning tasks, homework). ASSISTments is not a system as compact as other ITSs, it is designed more as a tool where teachers and researchers can try their experiments and ideas about learning.

Dybuster Calcularis

Dybuster Calcularis [64] is a software which trains basic mathematical skills that are fundamental for the understanding of math. Dybuster Calcularis contains various educational games which do not just train mathematical operations such as addition or subtraction, but they also train number processing skills. The system especially focuses on learners with difficulties in mathematics such as dyslexia or dyscalculia [63]. The system operates with a very detailed domain model with very specific skills and prerequisites and uses the Bayesian approach to learner modeling.

Math Garden

Math Garden [68] is an online environment for learners to practice math. Math Garden contains a broad range of exercises making it suitable for children from the age of four to middle school learners. Math Garden uses an Elo based model to estimate learners’ skill and
also, the response time information is incorporated into the model (more in Chapter 5).

Other systems

Three systems described above were the most important ones in the design of MatMat, but the internet offers many other related systems. We give a short description of some of them.

Khan Academy is well known for free, high-quality educational videos and other materials. It covers many educational domains, especially math. Khan Academy also offers a very comprehensive web-based system for training mathematical skills. The system contains a very detailed domain model which covers mathematics from counting to integrals. The system is based on the idea of mastery learning and learners train their skills by solving interactive exercises which offer hints and links to explanatory materials.

Carnegie Learning (an American publisher of math curricula for middle schools, high schools, and post-secondary learners) offers two web-based tutors for learning math: MATHia Software [113] aims at learners in grades 6 – 8 and Cognitive Tutor Software [5, 69] aims at learners in grades 9 – 12. Both systems work with an inner loop (step-by-step examples, contextualized hints, etc.) and use the concept of mastery learning. Both systems are very extensive and cover the targeted domain very well. Unfortunately, they are not available for free, and not all the research behind these systems is published.

Cognitive Tutor Authoring Tool (CTAT) [3] is a tool that supports the creation of ITS. With CTAT, it is relatively easy to create own tutors with the basic functionality of ITS without the necessity of building the system from scratch. One of the ITS developed with CTAT is Mathtutor [4]. The project includes a large number of problem sets to learn math by doing and contains detailed instructions and hints for all of the steps.

Among other systems with a strong use of adaptability or interesting practice tasks, there is ALEKS [55] which uses Knowledge space theory [41], FASTT Math with the focus on building fluency [71] or ST Math9 with thorough practice of fractions.

There are several math focused systems in the Czech environment, too. Probably the most used one is Matematika hrou\(^\text{10}\) which offers a wide range of exercises. The system offers support for teachers but completely lacks any type of adaptability. Another inspiring example is Matematika.in\(^\text{11}\) which is a system with a large number of original exercises based on the method of prof. Hejný [54]. These exercises practice math in a playful and practical manner.

### 2.5 Datasets

Here we give a basic description of the datasets used in this work.

#### 2.5.1 Real-World Data

All datasets have the same basic structure and consist of records of individual answers which are defined at the time when the answer was given, the id of the learner, id of the item, binary information of the correctness of the answer and response time. The rest of the information is specific for different datasets.

To reduce noise in the data, we usually filter out learners with less than 10 answers and items with less than 100 answers. These instances are not sufficient for a meaningful analysis of these learners or items, form only a small portion of the data and do not have a significant impact on further analysis.

Table 2.1 gives an overview of the used real-world datasets and their sizes.

**MatMat**

This dataset is smaller compared to other datasets but is valuable to this work due to the practiced domain in which response time information seems to be especially important. Part of the dataset is the item structure shown in Figure 2.6. This structure allows grouping items into concepts of various granularity or filtering out the answers related, for example, only to multiplication. We also give statistics for the five main subsets due to their often separate use.

11. [http://www.matika.in](http://www.matika.in)
2. Systems for Adaptive Practice

Table 2.1: Datasets overview. SR stands for the global success rate of the answers in a dataset.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>learners</th>
<th>items</th>
<th>answers</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>matmat</td>
<td>11344</td>
<td>1491</td>
<td>493340</td>
<td>0.834</td>
</tr>
<tr>
<td>- numbers</td>
<td>5993</td>
<td>60</td>
<td>104995</td>
<td>0.847</td>
</tr>
<tr>
<td>- addition</td>
<td>7473</td>
<td>598</td>
<td>123776</td>
<td>0.866</td>
</tr>
<tr>
<td>- subtraction</td>
<td>5629</td>
<td>280</td>
<td>67174</td>
<td>0.848</td>
</tr>
<tr>
<td>- multiplication</td>
<td>7812</td>
<td>429</td>
<td>125481</td>
<td>0.785</td>
</tr>
<tr>
<td>- division</td>
<td>5346</td>
<td>124</td>
<td>71914</td>
<td>0.832</td>
</tr>
<tr>
<td>geography</td>
<td>68079</td>
<td>1336</td>
<td>9935322</td>
<td>0.783</td>
</tr>
<tr>
<td>czech</td>
<td>40258</td>
<td>5557</td>
<td>15934541</td>
<td>0.867</td>
</tr>
<tr>
<td>anatomy</td>
<td>9313</td>
<td>2362</td>
<td>1110469</td>
<td>0.758</td>
</tr>
<tr>
<td>mathgarden-addition</td>
<td>41446</td>
<td>30</td>
<td>1800896</td>
<td>0.801</td>
</tr>
<tr>
<td>mathgarden-multiplication</td>
<td>58084</td>
<td>30</td>
<td>2898448</td>
<td>0.800</td>
</tr>
<tr>
<td>mathgarden-subtraction</td>
<td>40170</td>
<td>30</td>
<td>1653635</td>
<td>0.771</td>
</tr>
</tbody>
</table>

The most answers in MatMat are entered into a text input field, and thus all text answers are allowed. The rest of the answers are selected from a large pool of options, e.g., selection of a number and the number-line. The final answer is logged, and also individual steps are recorded and therefore it is possible to reconstruct how the answer was written (e.g., 1 → 12 → 1 → 13 → finish) and whether onscreen keyboard was used. A more detailed description of the datasets is available at [https://github.com/adaptive-learning/matmat-web/blob/master/data/data_description.md](https://github.com/adaptive-learning/matmat-web/blob/master/data/data_description.md).

Geography and Anatomy

The Geography dataset comes from the Outline maps system, where items correspond to places. To give a more detailed description of the questions, the dataset contains additional information about the type of question (choose place on the map or what is the highlighted place) and a list of the suggested options. Items can be divided by their type (states, rivers, cities, etc.) or by the map (World, Europe, Ger-
many, etc.). Together, these two classifications form contexts (e.g., European states) which correspond to the organization of the items in the user interface. The Anatomy dataset is analogous but comes from the Practice anatomy system. The data sets are described in [87] and at http://data.practiceanatomy.com/, respectively.

Czech

This data set comes from the Umíme Česky system. The dataset is the largest, but all answers have a 50% guess factor and thus carry the least information. The items are divided into concepts practicing specific grammatical phenomena. Some of the concepts also have manual labeling of the items corresponding to more granular grammatical rules. Unfortunately, for this dataset, response times are not available for most of the answers.

Math Garden

Data from Math Garden are partially publicly available as part of a previous research [22]. It has a similar domain focus to MatMat but is larger. Unfortunately, we have no detailed information about how the data was collected, which can hinder our interpretation of some results.

Data are divided into three sets corresponding to addition, multiplication and subtraction. In each dataset, only 30 most answered items were included. It is not possible to link users in these three datasets so we have to treat them separately.

Problem Solving Tutor

The dataset from the Problem Solving Tutor is the most specific. It does not contain information about the correctness of the answers but only response time, which is generally much larger than in other datasets. Due to the nature of the system, we call them response times rather solving times in this context.

Data are divided into subsets corresponding to the problem types in the system. The problem types vary considerably thus we treat them only separately, and Table 2.2 gives their overview. Note that the
individual items within a problem type can significantly differ thus the reported medians of solving times are only indicative and rather skewed because easier items are solved more often.

Table 2.2: Overview of used problems types in Problem Solving Tutor. ST stands for solving time.

<table>
<thead>
<tr>
<th></th>
<th>learners</th>
<th>items</th>
<th>answers</th>
<th>median ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary-crosswords</td>
<td>3778</td>
<td>58</td>
<td>71655</td>
<td>31s</td>
</tr>
<tr>
<td>nurikabe</td>
<td>350</td>
<td>54</td>
<td>6554</td>
<td>260s</td>
</tr>
<tr>
<td>region-puzzle</td>
<td>2194</td>
<td>130</td>
<td>38832</td>
<td>43s</td>
</tr>
<tr>
<td>robotanic</td>
<td>9399</td>
<td>79</td>
<td>127260</td>
<td>91s</td>
</tr>
<tr>
<td>rush-hour</td>
<td>6211</td>
<td>62</td>
<td>88493</td>
<td>23s</td>
</tr>
<tr>
<td>slitherlink</td>
<td>1068</td>
<td>94</td>
<td>18728</td>
<td>146s</td>
</tr>
<tr>
<td>sokoban</td>
<td>3745</td>
<td>82</td>
<td>35529</td>
<td>88s</td>
</tr>
<tr>
<td>title-maze</td>
<td>7313</td>
<td>112</td>
<td>121283</td>
<td>47s</td>
</tr>
</tbody>
</table>

2.5.2 Basic Statistics

The get a better grasp of the selected datasets we give (on top success rate in Table 2.1) some basic statistics. Figure 2.9 shows how fast learners are dropping out of the system. A steeper slope means that learners leave a system after fewer answers. The dataset with the most answers per user is Czech which is not surprising due to fastest practice. The other three datasets are from functionally very similar systems, but the geography dataset has more answers per learner. This is probably due to a wide use in schools (geography) and the relatively narrow focus of the topic (anatomy) or audience (matmat). We leave out mathgarden datasets because answers are filtered to 30 items a thus we cannot use them for this type of analysis. Note that the large periodic drops visible in the figure are caused by the set length of a practice set (10) after which it is a natural point to finish the practice session.

Due to the high importance of response times to this thesis, we also report the distribution of response times in Figure 2.10. As mentioned before, the Czech dataset is one with the shortest answers. The MatMat
Figure 2.9: Learner dropout. The graph shows how many learners stay in a system for (at least) a selected number of answers. The graph is normalized in respect to the number of learners. This type of graph can be used to easily read the median number of answers per learner.

dataset has the most varied times which supports the assumption of the high importance of response times in this dataset.

2.5.3 Simulated Data

For some analysis, we generated synthetic data. This approach has the advantage that we have background information such as the learner’s skill which is hidden in real-world scenarios. We can also study the influence of chosen parameters on results, etc.

For generating simulated data we use a simple approach with a minimal number of assumptions and ad hoc parameters. Each item belongs to one of $k$ concepts. Each concept component contains $n$ items. Each item has a difficulty generated from the standard normal distribution $d_i \sim \mathcal{N}(0, 1)$. Skills of learners with respect to individual concepts are independent. The skill of learner $l$ with respect to knowledge component $j$ is also generated from the standard normal distribution $\theta_{lj} \sim \mathcal{N}(0, 1)$. We assume no learning (constant skills).
Answers are generated as Bernoulli trials with the probability of a correct answer given by the logistic function of the difference of a relevant skill and item difficulty according to the Rasch model $p = \sigma(\theta_{lj} - d_i)$ described in section 4.1.1. This approach is rather standard, for example, Piech at al. [99] use a very similar procedure and other works also use closely related procedures [19, 45].

Datasets are generated for a different number of learners, concepts, and items in each concept. We denote these datasets for example as sim-100l-5c-20i for a dataset with 100 learners and 5 concepts with 20 items each, thus with $100 \times 5 \times 20 = 10 000$ answers.

This generation process can be altered for specific proposes to make data more similar to real-world scenarios. For example, we can remove a portion of the answers to make the data incomplete, skills of learners can be generated correlated or with a different variance.
3 Domain Modeling

Introduction

An important part of an intelligent system development is deep understanding of the knowledge domain system. We need to know which concepts the learner should master, what skill belongs to the knowledge domain, which task and other educational content are relevant for these skills and concepts, how this content should be structured and presented, etc. These questions are usually answered by domain experts and teachers, who have experience with the topic and teaching and understand the prerequisites and relationships between given concepts. However, this approach can be time-consuming, requires experts, hence it can be expensive and thus does not have to be available for a small developer group of intelligent systems. Moreover, even with the knowledge and experience of a domain expert, it is not always clear how to transfer this knowledge, useful in traditional education, to an online environment.

The goal of this research area is to provide methods which work with the collected data from the systems and provide tools and insight to human experts and make their work simpler and based on actual learner behavior, or to provide methods which describe a given knowledge domain automatically. We call this description the domain model and it can have various forms. In our setting, a domain model is a set of concepts, which cover specific parts of the knowledge domain and relationships between these concepts. We are especially interested in how a system’s educational content, which usually consists of a large pool of items, relates to these concepts.

A domain model has many applications in an adaptive system. It can be the foundation of a user interface, basis for meaningful feedback to learners and teachers or a part of an algorithm for the recommendation of concepts which should be practiced further. Moreover, the domain model describes which skills are necessary to track and thus it is an important input to a learner model and can have an impact on the behavior of an adaptive educational system. For example, the system can let the user practice items from one concept and after reaching mastery move to the next one. Another possible use of concepts is
3. Domain Modeling

for automatic construction of multiple-choice questions with good distractors (falling under the same concept). The building of such a domain model raises many questions. How many concepts do we need? What will these concepts be? How do we map our items to these concepts?

Outline

In this chapter, we describe a two-step approach to domain modeling as an alternative to the commonly used Q-matrix approach (Section 3.1.1). The first step is the computation of item similarities. In Section 3.2, we give an overview of possible similarity measures and compare them in diverse ways. The results presented in this work give guidance in the choice of an appropriate similarity measure for future research. We also introduce the second level of similarity, which is a novel approach that improves the properties of similarity measures, described in Section 3.2.2. On top of our publication [111] dedicated to similarity measures, we report more detailed evaluation results here.

The second step is a utilization of item similarities to a further construction or analysis of the domain model. This step offers vast possibilities and this work shows several directions of research. In Section 3.3 we show how item similarities can be used for automatic detection of concepts. We originally studied and published [16] this topic only in the context of solving times but, in this work, we also add an analysis of the correctness of answers. Section 3.4 describes a novel approach that uses learning data to improve a domain model provided by an expert, which was published in [83, 82]. Another use of item similarities is the visualization of a domain. Section 3.5 gives an overview of visualization techniques which mostly have not been previously published.

3.1 Related Work

3.1.1 Q-matrix

A usual approach to domain modeling in an educational context is the construction of a Q-matrix [11, 32, 123]. This model allows items to relate to multiple skills, which is typical for mathematics where solving
3. Domain Modeling

A problem often requires several skills. The mapping of the items to the concepts is expressed as a matrix (Q-matrix) where every row represents an item $i$ and every column represents a concept $c$. Value $q_{ic}$ then reflects the relationship of an item $i$ and a concept $c$ and it is usually binary (1 means that the item belongs to the concept) but can also be continuous and express the degree of the concept’s importance for the item. Many variants of this approach have been proposed under different names and terminology, e.g., matrix factorization techniques [30], sparse factor analysis [70], or matrix refinement [33].

The relationship between items and clusters is studied not only in educational data mining but also in a closely related area of recommender systems [107, 106]. The setting of recommender systems is in many aspects very similar to educational systems – in both cases we have users and items, just instead of performance (the correctness of answers, the speed of answers) recommender systems consider ratings (how much a user likes an item). Item similarities and clustering techniques have thus also been considered in the recommender systems research (we mention specific techniques below). There is a slight but important difference between the two areas. In recommender systems, item similarities and clusterings are typically only auxiliary techniques hidden within a ‘recommendation black box’. In an educational system, it is useful to make these results explicitly available to system developers, curriculum production teams, or teachers.

There are more types of the Q-matrix and their interpretations. The first possible interpretation is that only one of the concepts marked by a Q-matrix is necessary for the correct solution of the item (DINO, NIDO models [124]). But a more reasonable model is one that assumes that all the selected concepts are necessary for the correct solution (DINA, NIDA models [116]).

The basic idea of the construction of a Q-matrix automatically from data is to construct a simplified model that explains the observed data. Based on a response matrix of learners’ answers to items, we construct a model that predicts these answers. Typically, the model assigns several latent skills to learners and uses mapping of items to corresponding latent factors. This kind of models can often be naturally expressed using matrix multiplication, i.e., fitting a model leads to response matrix factorization of a skill matrix and a Q-matrix. An example of such a method is the alternate least-square factorization
3. Domain Modeling

method (ALS) [34], which uses alternation of the computation of skills from Q-matrix and responses and the computation of Q-matrix from skills and responses.

A part of the research is also focused on checking or improving the expert provided Q-matrix [27, 28, 35, 117]. The closest relevant work to our analysis in section 3.4 is [35], which used the matrix factorization approach for enhancing the expert provided matrix. Our setting is different, as we study only classification and we use problem-solving times instead of correctness data.

3.1.2 Clustering Methods

One of the possible domain models (simpler than Q-matrix) is a division of items into disjoint groups which represent concepts. The task is to use the data to find a division of items in such a way that items within concepts are more similar than those between them. This task fits the successful machine learning method called cluster analysis or clustering. There is a wide range of clustering approaches and methods [60, 59] and some of them were used successfully in an educational context [116]. Here we give a short description of some of those that we used in our analysis. The input for these algorithms is a list of points, a measure of similarity or distance (can be seen as dissimilarity) between these points and usually the number of expected clusters. The question of how to get the similarity measures from educational data is dealt with in the next section 3.2.

k-Means

The k-means algorithm is one of the most basic approaches, and it is an example of centroid based models. That means the points have to be in a metric space and clusters are represented by centroids which are also points in this space. A point belongs to the cluster with the closest centroid. Fitting of the centroids is done by alternating two steps. First, new clusters are created by assigning all of the points to the cluster with the closest centroid. Second, the position of all centroids is updated in such a way that centroids are means of all points in their clusters. In other words, the new centroid is a point with the minimal sum of squares of distances to all points in the cluster. The
steps are repeated until no change is produced or the maximum of steps is exceeded. The centroids can be initialized as random points, but it is necessary to make multiple runs of the whole procedure to find optimal clusters.

This algorithm also has a variant which does not need metric space but only similarities of the points. This method is called k-medoid and is analogous to k-means, but centroids can be only points obtained as input. A new centroid of the cluster is then found as the point with the smallest mean distance from other points in the cluster. The k-means has many limitations but also has many variants and extensions such as Gaussian mixture model fitted by the expectation-maximization algorithm. The k-means itself is a good baseline approach for its simplicity, small number of meta parameters (number of clusters, maximum of steps) and easy implementation.

Hierarchical Clustering

Hierarchical clustering does not search for a strict division of the items but rather builds a hierarchy of clusters. The hierarchy is represented by a binary tree where each node is a cluster of points, and its children are a division into two sub-clusters. The leaves are then single points. The advantage of this approach is that the number of clusters is not needed, and output information is more complex. However, this complex output is often harder to interpret and to pick relevant information. It is also possible to automatically get ordinary clusters if a number of required clusters is provided.

There are two main approaches to building this hierarchy. The divisive approach starts with the root with all points and in each step splits one cluster into two sub-clusters. The agglomerative approach starts with leaves and each step joins two nodes to their parent. The decision which clusters to join or split is based on linkage criteria which is a measure of the distance between two clusters. The most common criteria are

- complete linkage — maximum of distances between points in both clusters
- single linkage — minimum of distances
3. Domain Modeling

- **mean linkage** — mean of distances
- **ward method** — minimize the total within-cluster variance

**Spectral Clustering**

Spectral clustering is a less common approach than the two approaches described above and it is a technique based on linear algebra [81, 126, 134]. The central principle of the algorithm is the following. Firstly, a similarity graph for the data is created. This graph is based on the similarities of points and creation is basically pruning of the edges with a small similarity value. It is specific to the domain of application. Secondly, a Laplacian matrix of the similarity graph is constructed, normalized and its first \( n \) eigenvectors are computed, where \( n \) is the number of required clusters. Thirdly, the eigenvectors are used to transform original data into points in \( \mathbb{R}^n \), which are clustered using the standard k-means algorithm.

This approach deals with some problems of k-means, but it is at the cost of more meta-parameters which are necessary to be set correctly to get valuable results. The main advantage of this approach is that it can be used to get additional insight data. The \( \mathbb{R}^n \) space which we get as a byproduct of spectral clustering has dimensions sorted by importance according to clustering. For example, we can take the first two dimensions of this space and use them as a visualization of the originally high dimensional points.

### 3.1.3 Dimension Reduction Techniques

The general goal of these methods is to lower the dimension of high dimensional data. This can be used as a pre-step to other machine learning methods such as clustering, removal of redundant dimensions, data compression or data visualization. In our work, we mainly utilize data visualization techniques which are basically a projection or embedding into two dimensions which can be plotted. There is a wide variety of dimension reduction techniques [46]. Here we give a short description of the main approaches.
3. Domain Modeling

**Principal Component Analysis**

The Principal component analysis (PCA) is a baseline dimension reduction method. The idea behind PCA is to convert original points which are represented as possible correlated variables into a usually smaller number of uncorrelated variables called principal components. Furthermore, the principal components are ordered in such a way that the first one has the largest possible variance. In other words, PCA searches the subspace of the original space in such a way that the projection of points to the new subspace carries as much of the original information as possible.

The algorithm is straightforward, the basis of the new subspace is found as $k$, the first eigenvectors of the data covariance matrix. For a good result, some preprocessing of the data is necessary, mainly centering and normalization of all the variables. This method is easy to use and has a wide range of usage but the input data have to be represented as points in $\mathbb{R}^n$.

**Multidimensional Scaling**

Multidimensional scaling (MDS) aims for visualization more directly. The main idea behind MDS is to take the matrix of distances between points as input and to embed these points into a low dimensional space (usually a plane) in such a way that distances between points in this space are as close to the original distances as possible. To achieve this, MDS uses a stress function which describes the difference between actual and desired distances and tries to minimize this function. There are several stress functions which can be used based on the type of input data.

In contrast to PCA, for which output is only a linear transformation of input, MDS can work with more diverse types of input. The outputs are not deterministic and thus they can give fundamentally different results.

**t-SNE**

T-distributed stochastic neighbor embedding (t-SNE) [76] is a new and successful method specially designed for visualization of data. T-SNE has a similar basic idea to MDS: to embed points to lower dimensional
3. Domain Modeling

space in such a way that the original distance is somehow preserved. However, t-SNE does not work with distances directly but transforms them into probability distribution of the neighborhood over point pairs and uses an optimization procedure to preserve this distribution in the new embedding.

T-SNE has many good results across various fields and, in many cases, visualization using t-SNE is superior to other approaches and its outputs are usually much more human readable and interpretable. The downside of this approach is that a larger number of meta parameters have to be set specifically for the given data to get good results.
3. Domain Modeling

3.2 Similarity Measures

Most of the techniques we use to analyze or build domain models are based on information about items and especially on the similarity of items. The diagram in Figure 3.1 describes the general two-step process. Firstly, we compute the similarities between items represented as a symmetric item similarity matrix. Then, we use this matrix as input to a machine learning method which is chosen depending on the desired output. The advantage of this approach is that both steps are to some degree independent. If we have a different type of data or an alternative goal, we can just change how the similarity matrix is computed and leave the rest of the process the same. We can also combine several types of input, such as the learner’s answers, input from a human expert, semantic analysis of items, to a single similarity measure. On the other hand, when we have the similarity matrix, we can use it in more applications and get a different kind of results.

In this section, we deal with item similarities based on learner data and mainly on the correctness of answers. In case of data from the Problem Solving Tutor, we used solving times. In section 5.5.1, we give examples where correctness and time are combined. Note that in our discussion we use similarity measures (higher values correspond to higher similarity), but some applications require dissimilarity measures or distance of items (lower values correspond to higher similarity). This is just a technical issue, as we can easily transform similarity into dissimilarity by subtraction.

Figure 3.1: The diagram describes the two-step process of domain modeling techniques.
3. Domain Modeling

In our discussion, we mostly ignore the issue of learning (change of learners’ skill as they progress through items). When learning is relatively slow and items are presented in a randomized order, learning is just a reasonably small source of noise and does not have a fundamental impact on the computation of item similarities. We also use only the first answers of learners to each item so the learning effect at a particular item is minimized.

3.2.1 Similarity Measures Overview

In this section, we discuss different possibilities for the computation of item similarities from dichotomous data, i.e., correct or incorrect answers of learners to items. The input to item similarity computation is data about learner performance, i.e., a matrix $L \times I$, where $L$ is the number of learners and $I$ is the number of items. Values in the matrix specify learners’ performance: 0 for incorrect and 1 for correct. The matrix is typically very sparse (many missing values). Therefore, in computation of the similarity of an item pair, only the answers from learners who solved both items are considered. The output of the computation is an item similarity matrix $I \times I$, which specifies similarity for each pair of items.

With dichotomous data, we can summarize learners’ performance on items $i$ and $j$ using an agreement matrix with just four values (Table 3.1). Although we have just four values to quantify the similarity of items $i$ and $j$, previous research has identified a large number of different measures for dichotomous data and analyzed their relations [20, 45, 72]. For example, Choi et al. [20] discuss 76 different measures, albeit many of them are only slight variations on one theme. Similarity measures over dichotomous data are often used in biology (co-occurrence of species) [58]. A more directly relevant application is the use of similarity measures for recommendations [120]. Recommender systems typically use either the Pearson correlation or cosine similarity for the computation of item similarities [36], but they consider data richer than binary.

Table 3.2 provides definitions of 6 measures that we have chosen for our comparison. In accordance with previous research (e.g., [20, 58]), we call the measures by the names of researchers who proposed them. The choice of measures was made in such a way as to cover measures
Table 3.1: An agreement matrix for two items.

<table>
<thead>
<tr>
<th></th>
<th>item i</th>
<th></th>
<th>item j</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>incorrect</td>
<td></td>
<td>incorrect</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>correct</td>
<td>b</td>
<td>correct</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td>d</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: An overview of definitions of similarity measures based on the agreement matrix (\( n = a + b + c + d \) is the total number of observations).

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yule</td>
<td>( S_y = \frac{(ad - bc)}{(ad + bc)} )</td>
</tr>
<tr>
<td>Pearson</td>
<td>( S_p = \frac{(ad - bc)}{\sqrt{(a + b)(a + c)(b + d)(c + d)}} )</td>
</tr>
<tr>
<td>Cohen</td>
<td>( S_c = \frac{(P_o - P_e)}{(1 - P_e)} )</td>
</tr>
<tr>
<td></td>
<td>( P_o = \frac{(a + d)}{n} )</td>
</tr>
<tr>
<td></td>
<td>( P_e = \frac{((a + b)(a + c) + (b + d)(c + d))}{n^2} )</td>
</tr>
<tr>
<td>Sokal</td>
<td>( S_s = \frac{(a + d)}{(a + b + c + d)} )</td>
</tr>
<tr>
<td>Jaccard</td>
<td>( S_j = \frac{a}{(a + b + c)} )</td>
</tr>
<tr>
<td>Ochiai</td>
<td>( S_o = \frac{a}{\sqrt{(a + b)(a + c)}} )</td>
</tr>
</tbody>
</table>

used in the most closely related work and measures which achieved good results (even if the previous work was in other domains). We also tried to cover different types of measures.

The Pearson measure is the standard Pearson correlation coefficient evaluated over dichotomous data. In the context of dichotomous data, it is also called the Phi coefficient or Matthews correlation coefficient. Yule is a similar measure, which achieved good results in previous work [120]. The Cohen measure is typically used as a measure of inter-rater agreement (it is more commonly called Cohen’s kappa). In our setting, it makes sense to consider this measure when we view learners’ answers as ratings of items. Relations between these three measures are discussed in [138].
The Ochiai coefficient is typically used in biology [58]. It is also equivalent to cosine similarity evaluated over dichotomous data; cosine similarity is often used in recommender systems for computing item similarity, albeit typically over interval data [29]. The Sokal measure is also called Sokal-Michener or simple matching. It is equivalent to the accuracy measure used in information retrieval. Together with the Jaccard measure, they are often used in biology, but they have also been used for the clustering of educational data [45].

Note that some similarity measures are asymmetric with respect to 0 and 1 values. These measures are typically used in contexts where the interpretation of binary values is the presence/absence of a specific feature (or observation). In the educational context, it is more natural to use measures which treat correct and incorrect answers symmetrically. Nevertheless, for completeness, we have also included some of the commonly used asymmetric measures (Ochiai and Jaccard). In these cases, we focus on the incorrect answers (value $a$ as opposed to $d$) as these are typically less frequent and thus bear more information.

### 3.2.2 Second Level of Similarity Measure

The basic computation of item similarities computes the similarity of items $i$ and $j$ using only data about these two items. To improve a similarity measure, it is possible to employ a second level of item similarity that is based on the computed item similarity matrix and uses information on all items (see Figure 3.2). Note that the similarity measure used in the second level is computed with a different type of input, where items are represented by complete (no missing data) vectors of real numbers. Therefore it is not possible to use those measures described in the previous section which work only with dichotomous data, but only measures which can work with vectors of continuous values. Examples of such second step measures are Euclidean distance or Pearson correlation. The similarity of items $i$ and $j$ is given by the Euclidean distance or Pearson correlation of columns $i$ and $j$ in the original similarity matrix. Note that a second level measure may be used implicitly when we use a standard implementation of some clustering algorithms, e.g., $k$-means takes real vectors (columns of the similarity matrix) as input and uses Euclidean distance internally.
With the basic approach to item similarity, we consider items similar when the performance of learners on these items is similar. With the second step of item similarity, we consider two items similar when they behave similarly in respect to other items. The main reason for using this second step is the reduction of noise in the data by using more information. This may be particularly useful when dealing with learning. Two very similar items may have a rather low direct similarity because getting feedback on the first item can strongly influence the performance on the second item. However, we expect both items to have similar similarities to other items.

A more technical reason for using the second step (particularly the Euclidean distance) is to obtain a measure that is a distance metric. The measures described above mostly do not satisfy triangle inequality and thus do not satisfy the requirements on distance metric; this property may be important for some clustering algorithms. Note that the second level of the similarity measure can be applied repetitively, e.g., we can use Cohen, then Pearson in the second level and feed the result to k-means, which use extra Euclidean distance.

### 3.2.3 Comparison of Similarity Measures

In this section, we focus on the comparison of similarity measures, but we keep the overall context depicted in Figure 3.1 in mind. The quality of a similarity measure is only as good as the quality of the results of subsequent steps, therefore we consider several types of anal-

<table>
<thead>
<tr>
<th>matrix of answers</th>
<th>similarity matrix</th>
<th>similarity matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_1$</td>
<td>$i_m$</td>
<td>$i_n$</td>
</tr>
<tr>
<td>$l_1$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$l_2$</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>$l_3$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$l_L$</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 3.2: Schema describing the computation of a similarity matrix with the use of the second level of the similarity measure.
ysis. With simulated data, we analyze the similarity measures with respect to the ground truth while for real-world data we evaluate correlations among similarity measures. We also compare the quality of subsequent clusterings using an adjusted Rand index (ARI) [101, 133], which measures the agreement of two clusterings (with a correction for agreement due to chance). Typically, we use the adjusted Rand index to compare the clustering with the ground truth (available for simulated data) or with a manually provided classification. It can also be used to compare two detected clusterings (clusterings based on two different algorithms or clusterings based on two independent halves of data).

In evaluations with real-world data, we focus particularly on one subset of the czech dataset: questions about the choice between i/y in suffixes of Czech adjectives. For this subset, we have manually determined 7 groups of items corresponding to Czech grammar rules. We also used one more subset of the czech dataset (marked as Czech 2), mathgarden datasets and subsets of the matmat dataset (subsets defined by the type of mathematical operation), but these datasets have no manual subgroups.

**Correlation of Similarity Measures**

First, we analyze how similarity measures are related to each other. To evaluate how similar two measures are, we take all the similarity values for all item pairs and the computed correlation coefficient. Figure 3.3 shows the results for two data sets which are good representatives of the overall results. The Pearson and Cohen measures are highly correlated (> 0.98) across all data sets and have nearly the same values (although not exactly the same). Larger differences (but only up to 0.1) can be found typically when one of the values in the agreement matrix is small, which happens only for poorly correlated items with the resulting similarity value around 0. The second pair of highly correlated measures is Ochiai and Jaccard, which are both asymmetric with respect to the agreement matrix. The correlation between these two pairs of measures varies depending on the data set and in some cases drops by up to 0.5. Because of this high correlation within these pairs, we further report results only for the Pearson and Jaccard measures. The Yule measure is usually similar to the Pearson
3. Domain Modeling

Czech 1 (adjectives) MatMat: numbers

Figure 3.3: Correlations of similarity measures.

Figure 3.4 shows the effect of the second levels of item similarity on the Pearson measure (results for other measures are analogical). The Euclidian distance as second level similarity brings larger differences (lower correlation) than the Pearson correlation. The correlations for large data sets such as Math Garden are usually high (> 0.9) and, conversely, the lowest correlations are found in results for small data sets. This suggests that the second level of similarity is more significant, and thus potentially more useful, where only a limited amount of data is available.

Performance of Measures

As the first step in the evaluation of usefulness of the similarity measures, we consider experiments with a simulated dataset sim-100l-5c-20i (100 learners, 5 concepts with 20 items each) for which we can utilize the ground truth. In clustering, we expect high within-cluster similarity values and low between-cluster similarity values. Figure 3.5 shows a distribution of the similarity values for the selected measures.
and suggests which measures separate within-cluster and between-cluster values better and therefore which measures will be more useful in clustering. The results show that for the Jaccard and Sokal measures, the values overlap to a large degree, whereas the Pearson and Yule measures provide better separation.

Adding the second step – Pearson correlation in this instance – to the similarity measure separates within-cluster and between-cluster values better. This suggests that extending similarities in this way is not only a necessary step for some subsequent algorithms such as $k$-means but also a useful technique with better performance.

Finally, we evaluate the quality of the similarity measures according to the performance of the subsequent clustering. From the two considered clustering methods, we used hierarchical clustering in this comparison because it naturally works with the similarity measure and does not require metric space. The other two methods have similar results with the same conclusions. Table 3.3 and Figure 3.6 show the results. Although the results are dependent on the specific data
Figure 3.5: Differences between similarity values inside knowledge components and between them. A simulated data set with the basic setting was used.

set and the used clustering algorithm, there is a quite clear general conclusion. The Pearson and Yule measures provide better results than Jaccard and Sokal, i.e., the later two measures are not suitable for the considered task. Pearson is usually slightly better than Yule, but the choice between them does not seem to be fundamental (which is not surprising given that they are highly correlated). The results also show that the second step is always useful. The result for simulated data favors Euclidean distance over Pearson, but there are almost no differences for real-world data.
Table 3.3: Comparison of similarity measures for one real-world dataset and simulated data sets with 20 items per concept. The values are provided by the adjusted Rand index (with 0.95 confidence interval) for a hierarchical clustering computed based on the specific similarity measure. The top result for every data set is highlighted.

<table>
<thead>
<tr>
<th></th>
<th>Czech adjectives</th>
<th>sim-50l-5c</th>
<th>sim-100l-5c</th>
<th>sim-200l-5c</th>
<th>sim-100-2c</th>
<th>sim-100l-10c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson</td>
<td>0.32 ± 0.02</td>
<td>0.26 ± 0.04</td>
<td>0.48 ± 0.05</td>
<td>0.84 ± 0.05</td>
<td>0.77 ± 0.12</td>
<td>0.34 ± 0.04</td>
</tr>
<tr>
<td>Jaccard</td>
<td>0.31 ± 0.03</td>
<td>0.06 ± 0.03</td>
<td>0.15 ± 0.04</td>
<td>0.29 ± 0.08</td>
<td>0.32 ± 0.18</td>
<td>0.09 ± 0.02</td>
</tr>
<tr>
<td>Yule</td>
<td>0.31 ± 0.03</td>
<td>0.19 ± 0.04</td>
<td>0.43 ± 0.05</td>
<td>0.77 ± 0.07</td>
<td>0.60 ± 0.15</td>
<td>0.31 ± 0.03</td>
</tr>
<tr>
<td>Sokal</td>
<td>0.15 ± 0.06</td>
<td>0.11 ± 0.02</td>
<td>0.18 ± 0.03</td>
<td>0.25 ± 0.05</td>
<td>0.12 ± 0.11</td>
<td>0.14 ± 0.02</td>
</tr>
<tr>
<td>Pearson → Euclid</td>
<td><strong>0.43 ± 0.01</strong></td>
<td><strong>0.45 ± 0.05</strong></td>
<td><strong>0.80 ± 0.06</strong></td>
<td><strong>0.98 ± 0.01</strong></td>
<td><strong>0.95 ± 0.03</strong></td>
<td><strong>0.67 ± 0.04</strong></td>
</tr>
<tr>
<td>Jaccard → Euclid</td>
<td>0.31 ± 0.01</td>
<td>0.02 ± 0.01</td>
<td>0.01 ± 0.01</td>
<td>0.01 ± 0.01</td>
<td>0.05 ± 0.05</td>
<td>0.01 ± 0.00</td>
</tr>
<tr>
<td>Yule → Euclid</td>
<td>0.32 ± 0.02</td>
<td>0.36 ± 0.05</td>
<td>0.65 ± 0.07</td>
<td>0.94 ± 0.04</td>
<td>0.89 ± 0.11</td>
<td>0.43 ± 0.03</td>
</tr>
<tr>
<td>Pearson → Pearson</td>
<td>0.41 ± 0.03</td>
<td>0.39 ± 0.05</td>
<td>0.73 ± 0.06</td>
<td>0.96 ± 0.02</td>
<td>0.92 ± 0.03</td>
<td>0.55 ± 0.04</td>
</tr>
<tr>
<td>Jaccard → Pearson</td>
<td>0.38 ± 0.02</td>
<td>0.10 ± 0.04</td>
<td>0.18 ± 0.05</td>
<td>0.36 ± 0.11</td>
<td>0.58 ± 0.20</td>
<td>0.07 ± 0.02</td>
</tr>
<tr>
<td>Yule → Pearson</td>
<td>0.32 ± 0.03</td>
<td>0.38 ± 0.05</td>
<td>0.72 ± 0.06</td>
<td>0.97 ± 0.02</td>
<td>0.94 ± 0.04</td>
<td>0.55 ± 0.05</td>
</tr>
</tbody>
</table>
3. Domain Modeling

Figure 3.6: The quality of clustering for different measures used in the second step of item similarity. Top: Simulated data with 5 correlated skills. Bottom: Czech grammar with 7 manually determined clusters.
3. Domain Modeling

3.2.4 Do We Have Enough Data?

In machine learning, the amount of available data is often more important than the choice of a specific algorithm \[10]. Our results suggest that once we choose a suitable type of similarity measure (e.g., Pearson, Cohen, or Yule), the differences between these measures are not fundamental, the size of available data becomes a more important issue.

Specifically, for a given data set, we want to know whether the data are sufficiently large so that the computed item similarities are meaningful and stable. This issue can be explored by analyzing the confidence intervals for the computed similarity values. As a simple approach to the analysis of similarity stability, we propose the following approach: We split the available data into two independent halves (in a learner stratified manner). For each half, we compute item similarities and the correlation of the resulting item similarities.

We can also perform this computation for artificially reduced data sets – this shows how the stability of results increases with the size of data. Figure 3.7 shows this kind of analysis for our data (real-world data sets). We clearly see significant differences among individual data sets. The Math Garden data set contains a large number of answers and only a few items. The results show excellent stability, clearly, in this case, we have enough data to analyze item similarities. For the Czech grammar data set, we have a large number of answers, but these are divided among a relatively large number of items. The results show a reasonably good stability, the data are usable for analysis, but more data can bring improvement. For the MatMat dataset the stability is poor, we need more data to draw solid conclusions about item similarities.

3.2.5 Similarity Measures with Solving Times

In contrast with similarity measures computed from dichotomous data, with solving times we are limited to measures which also take continuous values as input. Note that we used logarithms of solving times as input, the result with unmodified data was consistently worse. In our comparison we used the Pearson correlation, Euclidean distance and Cosine similarity. Figure 3.8 shows a similar comparison to the
3. Domain Modeling

Figure 3.7: Stability of the similarity measure (Yule) for real-world data sets. The data set was sampled, split into halves and the Pearson correlation was computed for similarity values. Numbers on the right side indicate thousands of answers in data sets.

one shown in Figure 3.5, but here we used real-world data from the Problem Solving Tutor and problems were used as concepts. The figure shows the results for the pair of problems with a high number of answers, but the results are analogical for different problem pairs. The results show that the Pearson measure is superior to Euclid and Cosine and also the second level of similarity brings a significant improvement. Therefore, these results suggest that conclusions made with dichotomous data about the second level of similarity are also applicable to a different type of educational data. Also, the Pearson correlation coefficient seems to be universal and a good choice in various scenarios.
3. Domain Modeling

Figure 3.8: Differences between similarity values inside problems and between them. Data from the Problem Solving Tutor’s problems *Robotanic* and *Title Maze* was used.
3. Domain Modeling

3.3 Automatic Concept Detection

3.3.1 Clustering

In contrast to the Q-matrix approach, where an item can cover multiple concepts, we assume that each item belongs to exactly one concept. Therefore, the goal is to find such mapping \( l : I \rightarrow \Sigma \) of an item set \( I \) to a concept set \( \Sigma \). In our analysis, we use the clustering methods described in Section 3.1.2.

This model definitively has its limitations. It can describe the knowledge domain only as a division of items into concepts and it is not able to describe the degree of importance of a concept to a given item, relationships between concepts, etc. However, in the context of adaptive practice systems, where the items are mostly focused on a single knowledge or skill, this model seems to be a good fit. The simplicity of this approach is also its advantage. We do not need as much data as for more complex models, and results are easily interpretable. These properties are especially important in early phases of the development of a new system.

3.3.2 Algorithm Setting

In our comparison, we used rather standard settings of clustering algorithms, since alternative settings led to the same or worse results. For k-means, we used the best result (according to the internal cost function) from 100 runs with a maximum of 1000 iterations. For hierarchical clustering, we used the agglomerative approach and used complete linkage in case of item similarities used directly and ward method with Euclid distance used as the second level of similarity. For spectral clustering, we pruned the similarity graph to 10 nearest neighbors.

3.3.3 Evaluation

To compare two sets of clustering we used the adjusted Rand index (ARI) mentioned in Section 3.2.3. In some of our experiments, for a comparison of automatically detected concepts with the ground truth, we used the simple accuracy metric. In this approach, we first find the
best match for two sets of concepts and then accuracy is the ratio of
items placed in the same concept and all items.

The two metrics have a different approach to measuring similarity
of two clusterings; accuracy takes into account the placement of indi-
vidual points in contrast to ARI which works with point pairs. Thus
these metrics are not completely comparable, but in our experiments,
we did not come across an example in which one clustering is better
according to ARI and worse according to accuracy. Most of our results
are reported in ARI due to its frequent use and its easy interpretation:
0 ARI corresponds to random clustering and therefore no relevant
information. On the other hand, accuracy can be useful in scenarios
in which we are interested in the number of misclassified points.

To evaluate the quality of clustering, we have to choose a similarity
measure of items first. As can be seen in Table 3.3, Pearson as the
similarity measure with Euclidean distance as the second step is a
good choice in most scenarios. The clusters provided by clustering
algorithms were compared using an adjusted Rand index with the
ground truth for simulated data and with manual labels for real data.

Simulated Data

The results for the three clustering approaches are reported in Table 3.4.
We also evaluated different variants of clustering algorithms, e.g., it is a
common practice to perform dimension reduction before k-means, but
this approach always gives the same or worse results. From the three
main approaches, k-means and spectral clustering perform slightly
better than hierarchical clustering. However, hierarchical clustering
provides more complex information about items and does not require
the number of clusters ahead. This is an advantage in real world
scenarios where the number of concepts is typically unknown.

The results show that answers from 100 learners to 100 items are
sufficient to get good clusters. However, our simulated scenarios are
rather idealized. Figure 3.7 suggests that in real scenarios, millions
of answers are needed to get good clusters. It is probably because, in
reality, skills of learners are dependent, average difficulty of items is
shifted, and learners do not answer all items. All these factors can
influence the feasibility of finding the right clustering based on data.
Our experiments with simulated data confirm these expectations.
Table 3.4: Comparison of clustering approaches. The adjusted Rand index with 95% confidence interval is reported for one real data set and simulated data sets with the number of items fixed on 100 and various numbers of learners and knowledge concepts.

<table>
<thead>
<tr>
<th>learners</th>
<th>KC</th>
<th>k-means</th>
<th>hierarchical</th>
<th>spectral</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>2</td>
<td>0.87 ± 0.02</td>
<td>0.78 ± 0.03</td>
<td>0.85 ± 0.02</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>0.97 ± 0.01</td>
<td>0.94 ± 0.02</td>
<td>0.97 ± 0.01</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
<td>1.00 ± 0.00</td>
<td>1.00 ± 0.00</td>
<td>1.00 ± 0.00</td>
</tr>
<tr>
<td>50</td>
<td>5</td>
<td>0.65 ± 0.03</td>
<td>0.48 ± 0.03</td>
<td>0.57 ± 0.03</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>0.91 ± 0.01</td>
<td>0.79 ± 0.02</td>
<td>0.88 ± 0.02</td>
</tr>
<tr>
<td>200</td>
<td>5</td>
<td>0.99 ± 0.01</td>
<td>0.97 ± 0.01</td>
<td>0.99 ± 0.00</td>
</tr>
</tbody>
</table>

Czech adj. 7 0.42 0.38 0.45
Czech B 2 0.21 0.05 0.21
Czech L 2 0.08 0.00 0.04
Czech Z 2 0.76 0.88 0.82

The size of data is naturally a key factor (see Figure 3.6), but the results show that the density of data is also important. All answers from 100 learners to 100 items in 5 knowledge components are sufficient to obtain clusters with ARI 0.91. But if the same number of answers is spread between 200 or 500 learners, we get ARI only 0.81 or 0.31, respectively. Another important factor is orthogonality of concepts. In our basic version of simulated data, knowledge components are independent. In reality, skills on different concepts are correlated. When we generate skills of learners as correlated, we need significantly more answers to obtain quality clusters. To get ARI 0.9 with 2 concepts with a 0.7 skill correlation, we need answers from more than 1,000 learners. With a 0.85 correlation, the number of required learners is over 10,000. The last factor we investigated is the average difficulty of items. It is not surprising that the best result is obtained for 0 average difficulty, which corresponds to a 0.5 success rate and maximal information gain. For -1 average difficulty, which corresponds to a success rate larger than 0.73, the ARI decreases by up to 15%.
3. Domain Modeling

Dichotomous Data

The problem with real-world data is that we do not know the right answer. Therefore, we give some results based on manual labeling by a domain expert or concepts determined automatically from the features of the items.

Figure 3.15 shows the visualization of 4 subsets with different concepts represented by different colors. The visualization shows that the concepts — 7 manual ones for the adjective dataset and 2 concepts determined by the correct answer (i or y) for other datasets — are at least partially meaningful. Table 3.4 shows the results for these datasets. We can see that results vary across the datasets, for the L dataset, all methods failed but for the Z dataset, results show good concept detection. We are not able to say to what extent bad results are determined by bad clustering, insufficient data or bad ground truth concepts. However, the outcome of this analysis shows that the automatic approach can give relevant concepts and results are consistent with the results for simulated data (hierarchical clustering is slightly worse).

We give one specific example of clustering of items from the geography system. For this demonstration, we selected items which belong to the most practiced concept: European states. 3 clusters provided by k-means based on more than 800 thousand answers are the following:

- Czech Rep., France, Iceland, Italy, Germany, Poland, Austria, Russia, Slovakia, United Kingdom, Spain

- Albania, Bosnia and Herz., Bulgaria, Montenegro, Estonia, Lithuania, Latvia, Macedonia, Moldova, Serbia, Kosovo

- Belgium, Belarus, Denmark, Finland, Croatia, Ireland, Luxembourg, Hungary, Netherlands, Norway, Portugal, Romania, Greece, Slovenia, Sweden, Switzerland, Ukraine

The first cluster contains all states which are easily recognizable for Czech learners: neighboring states, islands and large states. The second cluster contains all the Baltic and Balkan states, and the last concept contains the rest of the states. In this example, the results are meaningful but not surprising. However, it shows that the automatic method
can give a useful insight for developers into a domain which is less clear than European states.

**Solving Time Data**

To evaluate concept detection from solving time data, we performed the following experiment. We mixed data from the Problem Solving Tutor for \( k \) different problems (e.g., two different logic puzzles) and removed information about item labels from the data. Then we let an algorithm analyze the data and cluster items into \( k \) groups. For the experiment, we used the 8 most solved problems from the Problem Solving Tutor: Sokoban, Slitherlink, Nurikabe, Binary crossword, Tilt maze, Robotanist, Rush hour, and Region puzzle. The experiment was performed on all \( k \)-subsets of these problems.

Figure 3.5 shows averaged results over all possible combinations of the problems for a given \( k \). Similarly to the results for dichotomous data, hierarchical clustering gives slightly worse results. Spectral clustering also outperforms \( k \)-means, but overall differences are rather small.

Table 3.5: Comparison of clustering approaches. For every number of concepts, \( k \) (problems) is the reported mean of adjusted Rand indexes of all possible combinations of problems.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( k )-means</th>
<th>hierarchical</th>
<th>spectral</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.60</td>
<td>0.48</td>
<td><strong>0.68</strong></td>
</tr>
<tr>
<td>3</td>
<td>0.51</td>
<td>0.41</td>
<td><strong>0.55</strong></td>
</tr>
<tr>
<td>5</td>
<td>0.36</td>
<td>0.33</td>
<td><strong>0.37</strong></td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>0.25</td>
<td><strong>0.28</strong></td>
</tr>
</tbody>
</table>

**3.4 Combining Data and Opinion of Expert**

In this section, we study how to combine the data-driven approach, specifically the similarity matrix, with input from a domain expert. In this setting, we have an expert labeling \( l_E : I \rightarrow \Sigma \) where \( I \) is the set of items and \( \Sigma \) is the set of skills. Such labeling may contain some
mistakes when compared to the correct hidden labeling $l$. The output of our algorithms is another labeling $l_A$ that may be different from $l_E$. The goal of our algorithms is to provide a more accurate labeling (according to $l$) than $l_E$.

In this work, we introduce how supervised classification methods can be used to detect errors in expert labeling. The main idea can be illustrated by the most straightforward approach which uses the $k$-NN ($k$-nearest neighbors) algorithm and similarity matrix. We assume that the most similar items belong to the same concept and thus have the same label. So for item $i_m$, a new label $l_A(i_m)$ will be the most common label among the $k$ most similar items from $I$ with problem $i_m$.

Of course, it is possible to use different classification methods. We have chosen the logistic regression classifier [42], which is more sophisticated but still simple to use. As input, we used columns from the similarity matrix, and Euclidean distance was implicitly used inside the classifier as the second level of similarity. The logistic regression classifier finds a linear combination of features of input data which – after passing through a logistic function – fits labeling the best. In fact, the classifier (for 2 concepts) finds a decision hyperplane which splits space into two parts corresponding to the concepts.

Figure 3.9 gives an inside look into how logistic regression works in our setting for 2 concepts. The classifier splits the space by a decision line into two half spaces which correspond to the components. Every point in the figure is a problem, and its shape represents labeling provided by an expert. White ones mark mistakes of the expert. For 3 or more concepts one-vs.-all strategy is applied.

The logistic regression classifier (similar as different classifiers) makes two types of mistakes in our training data:

1. Mistakes near the decision line, which are caused by a generalization and imperfect separability of problems.

2. Mistakes which correspond to the expert’s mistakes.

Our goal is to have a majority of type-2 mistakes and a minimum of type-1 mistakes. Because in $n$-dimensional space it is possible to split $n$ points by a hyperplane to all possible parts, the algorithm without regularization ($\lambda = 0$) overfits and never finds any mistakes. With
3. Domain Modeling

Figure 3.9: Projection of items onto the plane, for which the x-axis represents distance from the decision line. Data points represent problems with labeling provided by the expert (blue squares, red circles) and his actual mistakes (white ones).

growing regularization (we used a quadratic one), more mistakes of type 2 are found, but of course, some mistakes of type 1 occur.

3.4.1 Comparison of Approaches

We will denote the general approach described above as ED (expert-data). We compared this approach with the E (expert) approach, where only expert labeling is used and, the D (data) approach, where only the data is used in an unsupervised fashion.

Evaluation Setting

For the evaluation, we used data from 5 item types: Slitherlink, Tilt Maze, Rushhour, Sokoban, Robotanist. We chose these problems be-
cause they are some of the most popular ones and therefore a lot of data has been generated about the attempts to solve them.

To simulate multiple concepts for the evaluation purposes, we mixed data from $k$ item types together ($k \in \{2, 3, 4\}$). Each item type represents a single component (or label). An expert is simulated by taking the correct labeling and introducing random mistakes. Hence, in this situation (as opposed to the standard setting), we know the correct latent skills, and thus we can measure the accuracy of a method as the portion of the final labels assigned correctly.

Expert labeling was obtained as the correct labeling (given by problem types) with some additional mistakes. We performed these evaluations with different values of the expert error rate $p_e$ ranging from 0 to 0.5. When the simulated expert was labeling an individual data point, he assigned it the correct label with the probability of $1 - p_e$; otherwise, a random incorrect label was assigned. The expected accuracy of an expert with a $p_e$ error rate is $1 - p_e$. Expert labeling was then used in the ED approach as a training data set. As a data-driven D approach we used spectral clustering. For all approaches we used similarities computed by two levels of the Pearson correlation.

Algorithm Setting

In the case of the $k$-NN algorithm, experiments showed that the choice of $k$ is not essential; the results are comparable for $20 \leq k \leq \frac{|I|}{2}$. For the comparison we used a fixed $k = 30$.

For the method based on logistic regression, the choice of the regularization parameter $\lambda$ can be used to set the sensitivity of the algorithm to mistakes (see Figure 3.10). With low regularization, only the most serious (according to the data) mistakes are found (small recall and high precision). With high regularization, most of the mistakes of the expert are found (rising recall) together with some false positives (dropping precision).

Fortunately, the algorithm is not too sensitive to regularization (the figure uses the logarithmic scale of $\lambda$) and the interval of regularization with good accuracy is wide enough. So it is possible to use one regularization for different skills and gain good results. For a comparison, we used a fixed $\lambda = 2$. 

58
3. Domain Modeling

Figure 3.10: The effect of regularization on accuracy, recall and precision of finding the expert’s mistakes for the specific item type pair.

Evaluation

Figure 3.11 shows the comparison of the accuracies of the $E$, $ED$ and $D$ approaches. The top graph shows the results for the mix of 2 problems, the bottom one for the mix of 3 problems. We can denote three zones within the expert error rate based on which approach ($E$, $ED$ or $D$) performs the best. The $E$-zone appears for very small values of $p_e$, the $D$-zone appears for very large values of $p_e$, and the $ED$-zone appears between them, where the combination of expert labeling and learner data is beneficial.

We are interested particularly in the $ED$-zone, where the newly introduced approaches are the best, specifically in its position and width, which tells us for which values of $p_e$ these approaches are a good choice. Also, the maximal benefit of correcting expert labeling (over $E$ and $D$) can be measured. We will call this feature the height of ED-zone. Figure 3.12 shows ED-zone properties for various scenarios.

The figures show that the algorithm based on $k$-NN brings only a small improvement and only for a limited range of error rates. The ap-
3. Domain Modeling

Approach based on logistic regression is significantly better, and has very similar results to the approach based on the learner model with multidimensional skill published in the original paper [83]. This suggests that the results of these techniques are close to an optimal performance that is possible for the given data.

Figure 3.11: Comparison of techniques for particular situations (2 and 3 skills). The ED-zone is marked for the model.
3. Domain Modeling

Figure 3.12: ED-zones for the selected problem combinations. In each case, three techniques are compared: \( k \)-NN (K), logistic regression (L)). The degree of shade corresponds to the height of the ED-zone. The dotted lines mark points where spectral clustering has the same results as the simulated expert.

3.5 Visualizations Techniques

3.5.1 Similarity Graph

The most straightforward visualization of item similarities given by the similarity matrix is the similarity graph. Such a graph consists of nodes corresponding to items and edges connecting the most similar items. There are plenty of strategies to select these edges, and the right choice depends on the dataset and the desired result. In our analysis, we used edges for \( k \) highest similarities for each item independently, thus each item is connected to the graph even if it is dissimilar to all other items. One alternative is for example the selection of edges with a similarity higher than the threshold.
3. Domain Modeling

For an illustration of this approach, Figure 3.13 shows the selection of the most important similarities for European countries. Note that this graph contains some natural clusters such as Balkan countries (right), Scandinavian countries (middle), and well-known (by learners using our system) countries (left).

![Illustration of the networked model on European countries. Only the most important edges for each country are shown.](image)

3.5.2 Items Similarity Visualization

To visualize the similarities between items, we need to place these items into a two-dimensional space in such a way that distances between them on the plane correspond to values in the similarity matrix. To accomplish this, we use dimension reduction techniques described in Section 3.1.3. On top of that, we also used an alternative approach based on spectral clustering. One of the advantages of spectral clustering is the possibility to use the computed eigenvectors as a basis for dimension reduction. For a two-dimensional visualization, we can use the subspace defined by the second and the third eigenvector, which is also used in the k-mean step of the algorithm (the first eigenvector is a constant vector). Note that PCA in contrast to other techniques requires the placement of items into a Euclidean space (items are represented as columns in the similarity matrix) and therefore it forces the usage of Euclidean distance as the second level of item similarity.

We are not able to quantify the quality of visualization, so we give several examples of visualizations of various datasets with a short discussion.
Simulated Data

Figure 3.14 shows the comparison of different visualizations over simulated data. The standard approaches (PCA, MDS) successfully place the items from the same concepts close to each other, but concepts are difficult to be separated without the knowledge of the ground truth. The t-SNE provides much better results with visually separable clusters even for a large number of items where other approaches fail. Note that a suitable parameter setting of t-SNE (learning rate, perplexity, etc.) is important for the quality of the result, e.g., a bad choice of learning rate can easily lead to useless visualization. The alternative approach based on spectral clustering seems to get slightly better visualization than PCA and MDS, but it would still be difficult to identify the result without the knowledge of true labels.

Figure 3.14: Comparison of projections over simulated datasets sim-100l-5c-20i and sim-100l-5c-100i.

Czech Grammar

The top part of Figure 3.15 provides an example of a projection by t-SNE for a subset of items from the Czech grammar system. Items
3. **Domain Modeling**

are colored according to a manual labeling of items. In this specific case, the labeling process together with the visualization led to several actionable outcomes: reorganization of knowledge components within the system (due to a clear separation of items concerning different rules), addition of new items (with the goal of covering *empty spaces* in the visualization), and removal of several items (outliers in the visualization).

The bottom part shows other examples of projections by t-SNE. Here we visualize items focused on a specific grammar rule (writing *i/y* after a specific letter). For this concept, we do not have available manual labeling, so we colored items depending on the right answer. For the *B* and *Z* concepts, we can see a good separation of the two types of items, but for the *L* concept, which has the least answers per item, items are more mixed up. A closer look at individual items showed that close items are often very similar, e.g., a different form of the same word.

**Math**

Figure 3.16 shows a comparison of the visualization methods on two datasets from the mathematical systems. The Math Garden multiplication datasets are a selection of only 30 items, but we can see groups of similar items (e.g., multiplication by 1000, multiplication of smaller numbers, etc.). All the algorithms keep similar items close, but item groups are most visible with PCA and spectral clustering approaches. The MatMat dataset is much smaller, and we cannot see any significant groups. However, we used color to highlight a different type of number visualization used in items and we can see that items of the same type tend to be close.
Figure 3.15: Projection by t-SNE for a subsets of Czech grammar items with two options: i/y. Top: suffixes of adjectives. Colors correspond to manually determined concepts (grammatical rules). Shapes correspond to the correct answer of questions; Bottom: writing i/y after a specific letter (b, l and z).
Figure 3.16: Comparison of visualizations methods for datasets from Math systems. Top: multiplication items from Math Garden; Bottom: number practice items, color is determined by graphical representation used to visualize the number.
Problem Solving Tutor

The last example is a visualization of the Problem Solving Tutor data, specifically a projection of items from the problem type called Binary Crosswords. The goal in this problem is to fill a grid with zeros and ones in such a way that all specified conditions are met (Figure 3.17). This setting can be used for easy problems for practicing the basics of binary numbers (a) and logic operations (b), but also for more challenging problems where the specified conditions are given in a self-referential crossword manner, which leads to quite an entertaining practice of binary numbers and logic operations (c, d).

![Examples of Binary crossword problems.](image)

There are 55 problem examples in Binary Crosswords, and they can be easily divided into three main groups: examples based on the knowledge of the binary notation, examples which use logical operations, and the self-referential crossword examples which usually combine more different types of conditions and require deeper thinking. The Problem Solving Tutor contains a manually created classification of problems into these three types (binary numbers, logic operations, and crosswords). We apply the spectral clustering method to compute 3 clusters using the computed eigenvectors to plot items. In Figure 3.18, we can see that examples from a, b, c from Figure 3.17, which are typical examples of their groups, are placed in the middle of their clusters. Example d has a crossword form but strongly uses the concept of logic operations and location of this example corresponds to this observation. Another similar example is a circle in a
3. Domain Modeling

Binary number cluster which also has a crossword form, but solving it requires only the ability to write binary numbers. Interesting results were obtained for examples based on addition and subtraction of binary numbers. These examples were labelled as *binary numbers*, but 3.18 suggests that these examples are slightly different than other binary number examples and are closer to the *logic operations* cluster.

![Figure 3.18: Projection of all Binary crossword items onto a plane by spectral clustering with 3 algorithmically determined clusters.](image-url)

Figure 3.18: Projection of all Binary crossword items onto a plane by spectral clustering with 3 algorithmically determined clusters.
4 Learner Modeling

Introduction

As mentioned above, the understanding of the characteristics of individual learners is an essential condition for the adaptive behavior of a system. The goal of learner modeling [31] is to create an appropriate model which provides relevant information about learners and predicts their behavior. There are several aspects of a learner that can be modeled: their skill, speed, uncertainty, their affective or motivational state, etc. These usually hidden features have various observable manifestations from simple ones such as the correctness of answers or response time to complex ones such as eye movement or facial expression.

In this chapter, we focus on the modeling of learners’ skill based on the correctness of their answers which plays a key role in the adaptivity of most educational systems. In Chapter 5, we describe how this basic approach can be adapted or extended by the use of response times.

A learner model takes historical performances of learners (responses) as input. As mentioned, we focus on binary responses – correct or incorrect answers, but there are also other forms of responses such as points, partial credits or solving time. These data are used to train the model and consist of responses in the training set or all historical data available, depending on the type and usage of the model. Some models also require additional input such as information about items and concepts which is provided by a domain expert or by an automatic method (some of them described in Chapter 3).

For a given item and learner, the expected output is a prediction of the learner’s response on the item. In case of binary responses, the prediction is the probability of the correct answer. Moreover, models usually use internal parameters which allow them to estimate learners and item features and these parameters can also be of great use. A typical example is an estimation of learners’ skills, learning rate or item difficulty. All this output can be used for various ends such as detecting of mastery [13], item selection, content recommendation [121], providing feedback for learners or insight for developers and domain experts.
4. Learner Modeling

Outline

In our work, we focus on learner models based on the Elo rating system described in Section 4.1.4. A basic Elo based model was successfully used in adaptive practice systems with promising results. In this chapter, we introduce advanced models which deal with limitations of the basic model. In Section 4.2, we introduced 2 novel models for prior knowledge estimation: a concept and a network model. In Section 4.3, we introduce a hierarchical model which is tailored for the mathematics domain. All models have been published (the concept and the network model [84, 95], the hierarchical model [108, 109]) but in this thesis (Section 4.4) we give a more thorough evaluation on more datasets.

4.1 Related Work

In this section, we give an overview of the main modeling approaches. Two most related groups of models are models based on the logistic function and BKT model and its variants. We also describe the Elo rating system and learner models based on this system. These models are technically logistic models and should be a part of Section 4.1.1. However, Elo based models are essential for this work, thus we describe them separately in 4.1.4.

There are also other novel learner modeling approaches, such as models based on neural networks (deep knowledge learning) [67] or mixture modeling [122], but they are beyond the scope of this thesis.

4.1.1 Logistic Models

The following models represent learner skill as a continuous variable and typically also include the difficulty of an item also modeled as a continuous variable. The logistic function

\[ \sigma(x) = \frac{1}{1 + e^{-x}}. \]

serves as a link between a skill (or difference of skill and difficulty) and the probability that a learner answers correctly (see Figure 4.1).
In the case of multiple-choice questions, the probability is expressed by a shifted logistic function

\[ \sigma(x, k) = \frac{1}{k} + \frac{1 - \frac{1}{k}}{1 + e^{-x}} \]

The parameter \( k \) is the number of options and this expression is based on the assumption that minimal probability of success (reachable by a pure guess) is always \( \frac{1}{k} \).

Figure 4.1: Logistic relationship between skill and probability of success. Probability of success rises with an increasing skill of the learner.

**Performance Factor Analysis**

The simplest modeling approach based on the logistic function is the Performance factor analysis [51, 92], which is based on the Learning factor analysis [17]. The main idea is to start with an initial skill and change this skill value after every answer by a constant based on the correctness of the answer. The model naturally works with multiple skills and defines a separate set of parameters for each concept. For a given item \( i \), which belongs to a set of concepts \( C_i \), the learner skill is defined as follows

\[
\sum_{c \in C_i} \beta_c + \gamma c s_{lc} + \rho_c f_{lc}
\]

where \( \beta_c \) is the initial value of the skill and \( s_{lc}, f_{lc} \) track the number of successes and failures of the learner \( l \) using skill corresponding to the concept \( c \). Parameters \( \gamma_c \) and \( \rho_c \) express to what extent successes and failures testify about the skill value and \( \gamma_c \) is typically positive and \( \rho_c \) is negative. These parameters can be provided by an expert, based on the
4. Learner Modeling

Q-matrix described in section 3.1.1 or estimated automatically from data. The data-driven parameter estimation is a logistic regression problem which can be solved by maximum likelihood estimation or by brute force for a smaller number of skills.

From another point of view, PFA can be understood as a model which sets skills to $\beta$s for a new learner and updates all involved skills by $\gamma_c$ (if correct) or $\rho_c$ (if incorrect) after every learner’s answer.

Rasch Model

The Rasch model [139, 15] was originally introduced by Georg Rasch as a psychometric model. The model can be seen as a special simplified case of IRT; thus we give only a short description.

The model assumes one parameter for every learner $l$ – a skill $\theta_l$, and one parameter for every item $i$ – a difficulty $d_i$. The probability of success on item $i$ is then given by

$$P(\text{correct}, \theta_l) = \sigma(\theta_l - d_i) = \frac{1}{1 + e^{-(\theta_l - d_i)}}.$$  

The formula implies that the probability of success increases with increasing skill and decreasing difficulty. The difficulty of an item can be interpreted as a skill necessary to have a 50% chance of a correct answer.

IRT

Item response theory (IRT) [6, 52, 73, 74] has an origin in psychometrics in the 1950s. A classic application of IRT is a computerized adaptive testing which aims to an efficient discovery of skill (a latent trait in the original work). This goal (estimation of constant skills of a learner) is different than the goal of ITS (improve skills). Despite this fact, IRT proved suitable for tasks related to ITS. In the ITS context, the IRT model estimates the probability of the correct answer of a learner on an item. Therefore, the basic IRT model is suitable only in settings with items with binary (correct or incorrect) responses. There are many variants of IRT models and most of them differ in the number and type of parameters (features of learners and items). However, all IRT
models assume a logistic relationship between learner skill and the probability of success.

The simplest version of an IRT model is the one-parameter logistic model (1PL) which is equivalent to the Rasch model described above. A very popular extension of the 1PL model is the three-parameter logistic model (3PL) which adds two extra parameters to an item: the discrimination \( a_i \) and the pseudo-guessing parameter \( c_i \). The discrimination parameter \( a_i \) describes how much probability of success is sensitive to the skill of the learner. If \( a_i \) is high, only a small difference of the skill can significantly change performance. On the other hand, if \( a_i \) is close to zero, the performance on the item is almost independent of the skill. Parameter \( c_i \) gives a lower possible probability of success. For items with an open-ended answer, \( c_i \) is 0 but the parameter can be useful for multiple choice items where typically \( c_i = \frac{1}{n} \) for the number of choices \( n \). The overall model is described by the formula

\[
P(\text{correct}, \theta_l) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta_l - d_i)}}
\]

and Figure 4.2 gives better insight into the meaning of the parameters.

The scales of parameters are not uniquely determined and it is possible to transform parameters (i.e., add constant \( k \) to all skills and difficulties) without changing the predictive properties of the model.
This ambiguity is typically resolved by some normalization, e.g., requiring that the mean value of skill over the learners is 0 and variance is 1.

It is possible to extend IRT models with other parameters. One useful extension is considering a multidimensional skill \([1, 104, 105]\). In this case the term \(a_i(\theta_l - d_i)\) (which can be equivalently written as \(a_i\theta_l - d'_i\)) is replaced by \(\vec{a}_i\vec{\theta}_l - d'_i\). The term \(\vec{\theta}_l\) is a vector of skills of the learner and \(\vec{a}_i\) is a vector of the corresponding discrimination. This vector codes which skills are not necessary for solving this item (small or zero values) and which skills are needed (high values).

It is necessary to estimate all the parameters to apply these models in an ITS. Some of them are known (pseudo-guessing parameter \(c\)), some can be provided by a human expert (e.g., item difficulty). However, many parameters are unknown (e.g., learner skills) and input from the human expert can be inaccurate. Maximum likelihood estimation \([7, 14]\) is usually used to automatically estimate parameters from data. It is an iterative process which alternates between estimating item parameters from skills and skills from item parameters. This process can be initialized by zero or random values and it is ended when parameter values converge. However, this process is time-consuming and can be impractical in the online environment of an ITS.

### 4.1.2 Bayesian Knowledge Tracing

IRT was originally designed for testing tasks and estimating latent features, therefore it does not implicitly work with learning. In contrast to IRT, Bayesian knowledge tracing (BKT) \([23, 24, 128]\) assumes a change of skill over time, and learning plays an important role in the model.

The BKT model can be understood as a simple Bayesian network with two states that represent whether a learner has a skill or not. This reduction of skills to only binary information is a simplification of reality and therefore BKT is more suitable for very specific skills. Same as for IRT, binary responses (correct or incorrect answer) are expected. Another assumption is that every time a learner has an opportunity to use the skill (when solving an item) there is a chance of learning that skill.
There is a schematic description of BKT model in Figure 4.3 and the model has 4 parameters for every skill:

- \( P(L_0) \) initial skill – the probability that the skill is already known before the skill is used,
- \( P(T) \) learning – the probability that the skill will be learned when the skill is used,
- \( P(S) \) slip – the probability of an incorrect answer when the skill is known,
- \( P(G) \) guess – the probability of a correct answer when the skill is not known.

Then probability of knowing the skill after \( n \) relevant items and the probability of a correct answer are

\[
P(L_n) = P(L_{n-1}) + (1 - P(L_{n-1}))P(T) \\
P(\text{correct}) = P(L_n)(1 - P(S)) + (1 - P(L_n))P(G)
\]

The model uses conditional probabilities to update \( P(L_n) \) and, based on answers, provides more accurate predictions.

There are a lot of techniques that fit the parameters of the model. The most important ones include grid search (brute force method) [91, 9], expectation maximization [18], curve-fitting [24] and others.
are also a lot of extensions of the BKT model which, for example, incorporate forgetting [137, learner priors [90, 91], contextualization of guess and slip [9] or combine it with IRT [56].

4.1.3 Elo rating system

The Elo rating system [40] was created by Arpad Elo to estimate the skills of chess players. The main idea behind the Elo system is to update the skills of players after a game based on a difference between the expected result of the game and the actual result. If the result of the game is as expected (the better player won), the update is small and when the result is unexpected, the update is more significant.

The expected value (result) of the game is expressed by the probability of a win of the first player with skill $\theta_1$ over the second player with skill $\theta_2$ (similarly as in IRT, a logistic function is used):

$$P(E) = \sigma(\theta_1 - \theta_2) = \frac{1}{1 + e^{\theta_1 - \theta_2}}.$$  

The result of game $R$ would have value 1 if the first player won or value 0 if otherwise. The update of the skill after the game is the following:

$$\theta_1 := \theta_1 + K(R - P(E))$$

$$\theta_2 := \theta_2 + K(P(E) - R)$$

where $K$ is a suitably chosen constant.

An important extension of Elo is the Glicko rating system [49, 50] invented by Mark Glickman. The main contribution of Glicko to the measurement is an introduction of modeling uncertainty or ratings reliability which deals with the undesirable influence of an uncertain rating to other ratings. Rating reliability is used in the update of a rating, and it is decreased with every game and increased with time.

Apart from the rating of players of chess and other games, the Elo system has many applications including education [68, 88]. There are not two players competing against each other in an educational context but we can consider the learner as one player and an item as the second one. The skill of the item is interpreted as the difficulty of the item. The skill of the learner and the difficulty of the item are updated after every answer. Constant $K$ is usually different for the
4. Learner Modeling

Learner and item and it can be replaced by a function of variables such as response time [68], an order of answer to the item, etc.

4.1.4 Elo Based Learner Models

A common approach to the parameter estimation for the Rasch or IRT model is the joint maximum likelihood estimation (JMLE). This is an iterative approach that is slow for large data. The pre-train parameters — skills and difficulties — are then used but cannot be updated according to new learner data without running the whole estimation process again. This is not practical in a real-world system. It is particularly not suitable for an online application, where we need to adjust the estimates of parameters continuously.

Models such as BKT or PFA update information about a user after every answer: BKT updates the probability of being in the known state and PFA the skill of the user. Therefore, these models are more suitable for adaptive learning. However, also these models have a lot of parameters (e.g., information about contents such as difficulties) that are fixed, and it is necessary to provide them beforehand from training data.

In this chapter, we restrict our attention mainly to online models: models that update their parameters after each answer. Such models can adapt to user behavior quickly and therefore are very useful in adaptive practice systems. Previous work [89, 93] has introduced a variant of the Elo rating system called PFAE – PFA Elo/Extended, which allows estimating both skills and difficulties online and requires only a few meta-parameters beforehand.

This model has two main parts. The first part is basically an estimation of the Rasch model parameters and it is responsible for the estimation of prior knowledge (one dimension skill of learners) and difficulty of items. This part is described in the next section as the Basic model, and it is responsible for the processing of the first answers for each learner item pair. The second part is responsible for the estimation of current knowledge. Current knowledge tracks learner’s level of mastery for all individual items. Current knowledge is based on the estimation of prior knowledge and difficulties but is updated with repeated answers on the same item and thus can capture learning.
4. Learner Modeling

This part is described in more detail in section 4.3. The whole model is also described in the original work [89].

The PFAE model can be extended in various ways to work with more detailed information about answers, learners or items. The basic variant of PFAE does not use any domain model information. The extensions which use a domain model are described in the following Sections 4.2 and 4.3. Other extensions can incorporate the response time information (described in detail in Chapter 5) or more detailed information about wrong answers described in our publication [96].

4.2 Prior Knowledge Estimation

Modeling of prior knowledge was studied in our previous work [89, 90], but it gets relatively little attention. It is, however, very important, particularly in areas where learners are expected to have nontrivial and highly varying prior knowledge, which is the case of our adaptive practice systems that are often accessed from an internet search by diverse users. The estimate of prior knowledge is used in models of current knowledge (learning), i.e., it has an important impact on the ability of the practice system to ask suitable questions. We consider several approaches to modeling prior knowledge and explore their trade-offs.

Our aim is to estimate the probability that a learner $l$ knows an item $i$ based on previous answers of learner $l$ to questions about different items and previous answers of other learners to questions about item $i$. As a simplification, we use only the first answer about each item for each learner.

4.2.1 Basic Model

The basic model (described in previous work [89] and currently used in the online application) uses the key assumption that both learners and studied facts are homogeneous. It assumes that learners’ prior knowledge in the domain can be modeled by a one-dimensional parameter.

We model the prior knowledge by the Rasch model, i.e., we have a learner parameter $\theta_l$ corresponding to the global knowledge of a
learner $l$ of a domain and an item parameter $d_i$ corresponding to the difficulty of an item $i$. The probability that the learner answers correctly is estimated using a logistic function of a difference between the global skill and the difficulty:

$$P(\text{correct}|\theta_l, d_i) = \sigma(\theta_l - d_i).$$

The skill and difficulty estimates are updated as follows:

$$\theta_l := \theta_l + U(n) \cdot (\text{correct} - P(\text{correct}|\theta_l, d_i))$$
$$d_i := d_i + U(n) \cdot (P(\text{correct}|\theta_l, d_i) - \text{correct})$$

where $\text{correct}$ denotes whether the question was answered correctly and $K$ is a constant specifying the sensitivity of the estimate to the last attempt. An intuitive improvement, which is used in most Elo extensions, is to use an uncertainty function instead of a constant $K$ – the update should get smaller as we have more data about a learner or an item. We use an uncertainty function $U(n) = \alpha / (1 + \beta n)$, where $n$ is the number of previous updates to the estimated parameters and $\alpha, \beta$ are meta-parameters.

### 4.2.2 Concept Model

In the next model, which we call concept or simple hierarchical model, we try to capture the structure of the domain. The idea behind this model is to represent a user skill in more detail and make the model less sensitive to the assumption of homogeneity among learners. However, to use the concept model, we need to determine concepts (mapping of items into groups). This can be done in several ways. Concepts may be specified manually by a domain expert. In the case of the geography or anatomy learning application, some groupings are natural (continents, type of places, organ systems). It is also possible to create concepts automatically or to refine the expert provided concepts with the use of machine learning techniques described in Section 3.3.

In addition to the global skill $\theta_l$, the concept model also uses concept skills $\theta_{lc}$. We use an extension of the Elo system to estimate the model parameters. Predictions are done in the same way as in the basic Elo system, we just correct the global skill by the concept skill:

$$P(\text{correct}|\theta_l, \theta_{lc}, d_i) = \sigma((\theta_l + \theta_{lc}) - d_i).$$
4. Learner Modeling

The update of parameters is also analogical ($U$ is the uncertainty function and $\gamma$ is a meta-parameter specifying the sensitivity of the model to concepts):

$$
\theta_l := \theta_l + U(n_l) \cdot (\text{correct} - P(\text{correct}|\theta_l, \theta_{lc}, d_i))
$$

$$
\theta_{lc} := \theta_{lc} + \gamma \cdot U(n_{lc}) \cdot (\text{correct} - P(\text{correct}|\theta_l, \theta_{lc}, d_i))
$$

$$
d_i := d_i + U(n_i) \cdot (P(\text{correct}|\theta_l, \theta_{lc}, d_i) - \text{correct})
$$

This proposed model is related to several learner modeling approaches. It can be viewed as a simplified Bayesian network model [21, 66, 79]. In a proper Bayesian network model, we would model skills by a probability distribution and update the estimates using the Bayes rule; equations in our model correspond to a simplification of this computation using only point skill estimates. The Bayesian network model can also model more complex relationships (e.g., prerequisites), which are not necessary for our case. Related modeling approaches to the concept model are the Q-matrix method [11], which focuses on modeling mapping between skills and items (mainly using $N:M$ relations), and models based on the knowledge space theory [39]. Both these approaches are more complex than the proposed model. Our aim here is to evaluate whether even a simple concept based model is sensible for knowledge modeling.

4.2.3 Network Model

The concept model enforces a hard division of items into groups. With the next model, we bypass this division by modeling relations directly among individual items, i.e., we treat items as a network (hence the name network model). For each item, we have a local skill $\theta_{li}$. For each pair of items we compute the degree to which they are similar $c_{ij}$. All the similarity measures described in Section 3.2 can be used. This is done from the training data or – in the real system – once a certain number of answers is collected. After the answer to the item $i$, all skill estimates for all other items $j$ are updated based on $c_{ij}$. The model still uses the global skill $\theta_l$ and makes the final prediction based on the weighted combination of the global and local skill:

$$
P(\text{correct}|\theta_l, \theta_{li}) = \sigma(w_1\theta_l + w_2\theta_{li} - d_i).
$$
Parameters are updated as follows:

\[ \theta_l := \theta_l + U(n_l) \cdot (\text{correct} - P(\text{correct} | \theta_l, \theta_{li})) \]

\[ \theta_{sj} := \theta_{sj} + c_{ij} \cdot U(n_l) \cdot (\text{correct} - P(\text{correct} | \theta_l, \theta_{li})) \]

for all items \( j \)

\[ d_i := d_i + U(n_i) \cdot (P(\text{correct} | \theta_l, \theta_{li}) - \text{correct}) \]

This model is closely related to multivariate Elo which was previously proposed in the context of adaptive psychometric experiments [37]. Figure 3.13 from the previous chapter shows a selection of the most important correlations for European countries and gives an example of parameters \( c_{ij} \) used by the network model.

### 4.3 Current Knowledge Estimation

In the last section, we dealt with the estimation of prior knowledge, i.e., knowledge of learners before interacting with the system. Our high-level goal is to make systems for learning, and therefore we also have to take into account a change of the skill. We want to capture this change in our models, and we call this part of learner modeling the estimation of current knowledge.

Firstly, we give a description of the current knowledge estimation part of the PFAE model as described in [89], used in Outline maps and Anatom. Then, we propose a novel model called the hierarchical model which is more suitable for the domain of basic arithmetic.

#### 4.3.1 PFAE

The part of the PFAE model responsible for the estimation of current knowledge tracks parameters \( K_{li} \) for each learner \( l \) and item \( i \) he or she answered. This parameter captures condensed information about the learner’s knowledge of the item and the difficulty of this item. The probability of a correct answer is simply \( \sigma(K_{li}, n) \) for \( n \) options given in the question. The current knowledge parameter is initialized after the first answer of the learner to the item, and it is set to the difference of the learner’s prior knowledge and the difficulty of the item

\[ K_{li} := \theta_l - d_i. \]
4. Learner Modeling

Note that this first answer is primarily processed by the prior knowledge estimation part of the model.

This parameter is then updated after each answer of the learner to the item in the following way

\[
K_{li} := K_{li} + \gamma (1 - \sigma(K_{li}, n)) \quad \text{for a correct answer,}
\]

\[
K_{li} := K_{li} - \delta (\sigma(K_{li}, n)) \quad \text{for an incorrect answer.}
\]

The meta-parameters \(\gamma\) and \(\delta\) are inspired by PFA (whence the name PFAE) and describe the sensitivity of the current knowledge to correct and incorrect answers. The rest of the update equations is the *Elo part* and it is equivalent to the term \(\text{correct} - P(\text{correct} | K_{li})\) which expresses the unexpectedness of the response.

This model has some limitations. The first one is the assumption of an independence of items, i.e., learning one item has no impact on learning other items. This assumption seems to be reasonable in the practice of facts (e.g., geography places) where the memorization of one item has a small impact on the knowledge of other items. However, in other domains like mathematics, which is less about declarative knowledge and more about procedural knowledge and all items are more or less connected, this assumption does not hold.

The second limitation of this approach is that after the current knowledge parameter \(K_{li}\) is initialized, it is not updated according to new observations about the prior skill or the difficulty of the item. This effect is the most pronounced at the beginning of practice when the model has no information about the learner. As an example, consider an excellent learner who correctly answers his first item. After his answer, the \(K_{i1}\) is set based on the difficulty of the item and prior knowledge that is right now only slightly above zero (updated only once). After the following 20 answers about different items all answered correctly, the model correctly estimates a high prior knowledge but \(K_{i1}\) is still frozen at average knowledge and the learner has to answer this item a couple of times to get a good estimation. Similar problems but with the difficulty of items can be experienced by the first users in the system who answer when the difficulties are not estimated reliably yet.
4. Learner Modeling

4.3.2 Hierarchical Model

Already in the early stages of the development process of the MatMat system, it was clear that the assumption of item independence is not satisfied and thus the PFAE approach is not appropriate for the domain of basic arithmetic operations. The learning of one item such as $5 \times 6$ evidently influences other similar items like $4 \times 6$ or $5 \times 7$. Also, the mastery of the concept of multiplication can influence progress in the concept of division, etc. So we look for a learner model that can propagate information about learning between concepts.

In the context of MatMat, we also assume that the tracking of current skills at the level of individual items is too detailed information. For example, we can have only one skill for items $7 \times 8$ and $8 \times 7$ with various graphical representations. Another example is grouping all items in the concept of multiplication of numbers larger than 20 where tracking more detailed information does not seem reasonable. So, we seek a model that is able to group a various number of items under one skill but still tracks individual difficulties.

We proposed a model called the Hierarchical model which can be seen as the Concept model but with more levels of skills. This approach does not explicitly distinguish between prior and current knowledge either, and it estimates the skill of learners as is and updates this estimation when the skill is changed. We give a description of this learner model and the underlying domain model on a specific example of model for MatMat system but a similar approach can be used in other scenarios with domain specific changes.

Domain Model

Mathematics is a very complex domain full of diverse components and relationships. Even in our very simplified case, when we considered only the basics, the situation is still relatively complicated. One approach to building a domain model for mathematics is based on the Knowledge space theory [38]. This approach splits the curriculum into skills and defines relations of the prerequisites between them. This oriented graph can then be treated as a dynamic Bayesian network [65].

We used a different approach and organized the concepts (which also correspond to learners’ skills) into a tree structure depicted in
Figure 4.4. Each node corresponds to a concept and its successors to more specific sub-concepts. The similarity of concepts can be expressed as the level of the nearest common ancestor. Note the fact that a concept \( c \) is an ancestor of a concept or item \( i \) as \( c > i \).

![Diagram of the tree structure of the domain model used in MatMat.](image)

The root of the tree is a global skill which represents overall knowledge of mathematics. Under that there are skills which correspond to basic units in the system (level 1) — numbers, addition, subtraction, multiplication, and division. In level 2 are sub-skills which represent concepts (inspired by [64]) within the parent skill, e.g., under ‘numbers’ skill there are ‘numbers in range from 1 to 9’, ‘numbers in range from 10 to 20’, ‘numbers greater than 20’; or under ‘addition’ there is ‘addition in range from 1 to 9 (without bridging to 10)’, ‘addition in range from 10 to 20 with bridging to 10’, etc. And finally, level 3 skills correspond to the tasks for which a mastery on the level of declarative knowledge is expected. An example of these are skills that correspond to numbers (1, 2, 3, ...), simple addition tasks (1 + 2, 5 + 7) or multiplication of numbers smaller than 10 (3 · 5, 7 · 8). There are no level 3 skills for a more complicated task (e.g., 11 · 13) for which procedural knowledge is always involved. The items representing these tasks typically belong to the more general level 2 skills.

In the current model, every item in the system is mapped to exactly one skill (typically a leaf skill). So, there are multiple items under one skill. In case of the more general level 2 skills, it can be tens or...
hundreds. In case of the level 3 skills, there are from 2 to 10 items which are various forms of the task ($5 + 7$ and $7 + 5$) and different graphical representations of the task (numbers, objects, number line, etc.).

**Learner Model**

As mentioned above, this model can be seen as a more complex version of the concept model described in Section 4.2.2. The model estimates the difficulty $d_i$ of the item $i$ as usual and the skill $\theta_{ic}$ for each concept $c$ in the domain model and each learner $l$. The skill used in the estimation of the probability of a correct answer for the item $i$ is then given by the sum of all ancestor skills in the domain model. Therefore, the probability of a correct answer is

$$P(\text{correct} | \theta_l ?, d_i) = \sigma(\sum_{c \geq i} \theta_{lc} - d_i).$$

The update of parameters is a combination of the prior and current part of the PFAE model. The updated skills are those which are ancestors of an answered item, and they are updated proportionally to the correctness of the answer, the expected response (probability) and the number of previous learner’s answers in the concept. The concept skills are updated sequentially from the most global (math in the mathematics domain example) to the concepts in lower levels in such a way that after each update the expected response is re-computed with previously updated skill values. Note that an update of difficulty is done only if the item is answered by the learner for the first time.

$$\theta_{lc} := \theta_{lc} + K \cdot U(n_{lc}) \cdot (\text{correct} - P(\text{correct} | \theta_{lc}, d_i))$$

$$d_i := d_i + U(n_i) \cdot (P(\text{correct} | \theta_l, \theta_{lc}, d_i) - \text{correct})$$

$U(n) = \alpha / (1 + \beta n)$ is the uncertainty function described above ($n_{lc}$ is the number of the learner’s answers in the concept) which causes that the more global skills are updated more slowly because they are updated more often. The meta parameter $K$ depends on the correctness of the answer and determines the sensitivity of the model to correct (parameter $\gamma$) and incorrect answers (parameter $\delta$). Note that in the original version of the model, an update of a skill did not
use the uncertainty function $U$ but another constant dependent on the level of the skill, in such a way that the skill corresponding to a more specific and smaller concept was updated by a larger value than a skill corresponding to a larger concept.

4.4 Evaluation

As a baseline, we used 3 simple models. The first one is the *global average* and its predictions are given as the number of previous correct answers divided by the number of all previous answers, i.e. this model after a while always gives almost the same prediction independently of the learner or item, and it corresponds to the global success rate. The second model is called *item average* and it works similarly as the previous one, but it tracks the success rate for each item independently instead of one global value. The prediction for a question on an item is the current success rate on this item. Therefore, this model, in fact, works with the difficulty of the items but does not take into account the skills of the learners. The last model is *learner average* and it is analogical to *item average* but it tracks the success rates of the learners instead of the items.

4.4.1 Metrics

To evaluate the predictive power of the models, we have to choose a metric to compare predictions and expected values. There are several metrics that are commonly used for the comparison of learner models such as the mean absolute error (MAE), log-likelihood (LL), area under the curve (AUC) or root mean square error (RMSE). Previous research [94] argues for the use of RMSE, which we use primarily in our evaluation but we also report other relevant metrics.

RMSE is closely related to the Brier score [94], which provides a decomposition [80] into uncertainty (measures the inherent uncertainty in the observed data), reliability (measures how close the predictions are to the true probabilities) and resolution (measures how diverse the predictions are).

This decomposition can also be illustrated graphically. Figure 4.5 shows a comparison of the basic model and the concept model for...
the estimation of prior knowledge. Both calibration lines (which are near the optimal one) reflect very good reliability. On the other hand, histograms reflect the fact that the concept model gives more divergent predictions and thus has a better resolution.

Figure 4.5: Illustration of the Brier score decomposition for the basic model and the concept model. Top: reliability (calibration curves). Bottom: resolution (histograms of predicted values).
4. Learner Modeling

4.4.2 Prior Knowledge Model Evaluation

To evaluate prior knowledge models, we used the geography, anatomy and matmat datasets. All datasets were split into a train set (30%) and a test set (70%) in a learner-stratified manner. As a primary metric for model comparison and parameter fitting, we used the root mean square error (RMSE) since the application of the model in the systems works with absolute values of predictions. The train set was used for finding the values of the meta-parameters of individual models.

Model Parameters

Grid search was used to search the best parameters of the uncertainty function $U(n)$. The left part of Figure 4.6 shows RMSE of the basic model on training geography data for various choices of $\alpha$ and $\beta$. We chose $\alpha = 1$ and $\beta = 0.04$ for geography and anatomy ($\alpha = 1.2$ and $\beta = 0.04$ for matmat) and we used these values also for the derived models which use the uncertainty function. For the concept model we found the best $\gamma = 0.4$ ($\gamma = 0.5$ for matmat and $\gamma = 0.2$ for anatomy).

The grid search (Figure 4.6 right) was also used to find the best parameters $w_1 = 0.6$, $w_2 = 0.6$ ($w_1 = 1$, $w_2 = 1.5$ for matmat) of the network model. The train set was also used for the computation of similarities $c_{ij}$; we used double Pearson similarity. To avoid spurious high similarities of two items $i, j$ as the consequence of a lack of common learners we set all $c_{ij} = 0$ for those pairs $i, j$ with less than 200 common learners.

The concept model for our evaluation uses manually determined concepts:

- **geography** — concepts based on both location (e.g., continent) and type of place (e.g., country)

- **anatomy** — concepts based on organ system (e.g., muscles) and body part (e.g., head)

- **matmat** — level 1 concepts (e.g., multiplication)
4. Learner Modeling

Figure 4.6: Grid searches for the best uncertainty function parameters $\alpha, \beta$ (left) and the best parameters $w_1, w_2$ of the network model (right). As can be seen from the different scales, models are more sensitive to $\alpha$ and $\beta$ parameters. This example of a grid search is for the geography dataset.

Accuracy of Predictions

All the reported models work online, i.e., update their parameters ($\theta_l$ and $d_i$) after every answer. Training of models is started on the train set, but it continues on the test set. Only predictions on this set are used to evaluate models.

Table 4.1 shows the results of the model comparison with respect to RMSE, LL and AUC, but the main result is not dependent on the choice of metric. In fact, predictions for individual answers are highly correlated. For example, for the basic model and the concept model, most of the predictions (95%) differ by less than 0.1.

All Elo based models bring a large improvement over baseline models. Both the concept model and the network model bring an improvement over the basic model. The improvement is statistically significant (as determined by a t-test over the results of a repeated cross-validation), but it is rather small. We hypothesize that the improvement of the extension models will be more significant for less homogeneous populations of learners.
### 4. Learner Modeling

Table 4.1: Comparison of prior knowledge models for 3 datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE</th>
<th>LL</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>geography</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global avg.</td>
<td>0.45032</td>
<td>−1554613</td>
<td>0.5086</td>
</tr>
<tr>
<td>Item avg.</td>
<td>0.43660</td>
<td>−1474196</td>
<td>0.6487</td>
</tr>
<tr>
<td>Learner avg.</td>
<td>0.45466</td>
<td>−1867553</td>
<td>0.5984</td>
</tr>
<tr>
<td>Basic</td>
<td>0.41345</td>
<td>−1341613</td>
<td>0.7439</td>
</tr>
<tr>
<td>Concept</td>
<td><strong>0.41156</strong></td>
<td><strong>−1331157</strong></td>
<td><strong>0.7496</strong></td>
</tr>
<tr>
<td>Network</td>
<td>0.41232</td>
<td>−1334710</td>
<td>0.7475</td>
</tr>
<tr>
<td><strong>matmat</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global avg.</td>
<td>0.37162</td>
<td>−117657</td>
<td>0.5252</td>
</tr>
<tr>
<td>Item avg.</td>
<td>0.36643</td>
<td>−114280</td>
<td>0.6278</td>
</tr>
<tr>
<td>Learner avg.</td>
<td>0.36934</td>
<td>−154023</td>
<td>0.6713</td>
</tr>
<tr>
<td>Basic</td>
<td>0.34927</td>
<td>−103884</td>
<td>0.7372</td>
</tr>
<tr>
<td>Concept</td>
<td><strong>0.34613</strong></td>
<td><strong>−102349</strong></td>
<td><strong>0.7499</strong></td>
</tr>
<tr>
<td>Network</td>
<td>0.34928</td>
<td>−103866</td>
<td>0.7372</td>
</tr>
<tr>
<td><strong>anatomy</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global avg.</td>
<td>0.44160</td>
<td>−313551</td>
<td>0.5088</td>
</tr>
<tr>
<td>Item avg.</td>
<td>0.43411</td>
<td>−304720</td>
<td>0.6118</td>
</tr>
<tr>
<td>Learner avg.</td>
<td>0.44458</td>
<td>−357917</td>
<td>0.5943</td>
</tr>
<tr>
<td>Basic</td>
<td>0.41442</td>
<td>−281598</td>
<td>0.7145</td>
</tr>
<tr>
<td>Concept</td>
<td><strong>0.41392</strong></td>
<td><strong>−280938</strong></td>
<td><strong>0.7171</strong></td>
</tr>
<tr>
<td>Network</td>
<td>0.41399</td>
<td>−280979</td>
<td>0.7150</td>
</tr>
</tbody>
</table>
4. Learner Modeling

Table 4.2: Comparison of learner models on the *matmat* dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE</th>
<th>LL</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global avg.</td>
<td>0.37480</td>
<td>−154610</td>
<td>0.5292</td>
</tr>
<tr>
<td>Item avg.</td>
<td>0.36928</td>
<td>−149989</td>
<td>0.6292</td>
</tr>
<tr>
<td>Learner avg.</td>
<td>0.36342</td>
<td>−180677</td>
<td>0.7037</td>
</tr>
<tr>
<td>PFAE</td>
<td>0.33023</td>
<td>−122305</td>
<td>0.8141</td>
</tr>
<tr>
<td>Hierarchical</td>
<td><strong>0.31907</strong></td>
<td><strong>−117002</strong></td>
<td><strong>0.8181</strong></td>
</tr>
</tbody>
</table>

4.4.3 Hierarchical Model Evaluation

We compared the hierarchical model to the PFAE model described in Section 4.3.1 and the baseline models described at the beginning of Section 4.2, which can be straightforwardly used also in the prediction of current knowledge. For this evaluation, we used only the *matmat* dataset for which the hierarchical model was primarily designed.

For this comparison, we used the same $\alpha = 1.2$ and $\beta = 0.04$ as for the comparison of prior models and additionally we found optimal parameters $\gamma = 2$ and $\delta = 0.5$ for the PFAE model and $\gamma = 1$ and $\delta = 0.75$ for the hierarchical model. Table 4.2 shows the results for various metrics. Again, the results are not dependent on the choice of metric and PFAE, and the hierarchical model outperformed the baseline models. The hierarchical model also has significantly better results than the PFAE model. Moreover, this model brings more complex information about the learner’s skill which is practically used in feedback to learners and their teachers.

4.4.4 Using Models for Insight

In learner modeling, we are interested not just in predictions, but also in getting insight into the characteristics of the domain or learning process. The advantage of more complex models may lie in additional parameters, which bring or improve such insight.
4. Learner Modeling

**Difficulty of Items**

The difficulty of items gives insight into the domain, and it is important feedback for system developers about the content of the system. For example, too easy or too difficult items can be revised, or more items can be added to fill the gaps in difficulty.

Figure 4.7 (left) gives a comparison of item difficulty for the basic model and the concept model. As we can see, the estimated values of the difficulties are quite similar; thus more advanced models do not bring improvement in this direction. Figure 4.8 shows the difficulty development for selected items for training of the Basic model with the geography dataset. These results suggest that 500 answers are enough to get a good estimation of item difficulty and after 2000 answers the difficulty is stable. Note, that a difficulty can shift, for example, with a changing composition of the learner population.

![Figure 4.7](image.png)

**Learners’ Knowledge**

Models with multiple skills bring additional information not just about the domain, but also about learners. Figure 4.7 (right) shows estimated global skill parameters of the basic and the concept model. The skills
are slightly less correlated than the difficulty parameters, probably due to fewer answers per parameter, but the correlation is still very high.

The main difference of a more advanced model lies in the concept skills. The correlation of concept skills with the global skill ranges from -0.1 to 0.5; the most correlated concepts are the ones with a large number of answers like European countries (0.48) or Asian countries (0.4), since answers on items in these concepts also have a large influence on the global skill. The correlation between two clusters of skills typically ranges from -0.1 to 0.1. These low correlation values suggest that concept skills hold interesting additional information about learners’ knowledge.

**Domain Model Quality**

More complicated models (network, concept) bring insight into the domain thanks to the analysis of relations between items, e.g., by comparing various sets of item clusters which can be provided by a technique described in Chapter 3.

We performed an evaluation of the concept model with different concepts. We used several approaches for specifying the concepts man-
4. Learner Modeling

ually. For geography, we used concepts based on type (e.g., countries, cities, rivers), location (e.g., Europe, Africa, Asia) and combination of the two approaches (e.g., European countries, European cities, African countries). For anatomy, we used concepts based on the organ system (e.g., muscles, bones), location (e.g., head, arm) and also a combination of these.

Since we have most learners’ answers for European countries, we also considered a data set containing only answers on European countries. For this data set, we used two sets of concepts. The first one is the division into Eastern, Western, Northwestern, Southern, Central and Southeastern Europe, the second concept set is obtained from the first one by merging Central, Western and Southern Europe (countries from these regions are mostly well-known by our Czech learners) and merging Southeastern and Eastern Europe.

We compared these manually specified concepts with automatically corrected (method described in Section 3.4) and entirely automatically constructed concepts (described in Section 3.3). The quality of the concepts was evaluated using prediction accuracy of the concept model which uses these concepts. Table 4.3 shows the results expressed as RMSE improvement over the basic model. Note that the differences in RMSE are necessarily small since the used models are very similar and differ only in the allocation of items to concepts.

For the whole data set, a larger number of concepts brings an improvement of performance. The best results are achieved by manually specified concepts (combination of location and type of place), automatic correction does not lead to a significantly different performance. For the smaller data set of European countries (39 items), a larger number of (both manual and automatically determined) concepts brings a worse performance – a model with too small concepts suffers from a loss of information. In this case, the best result is achieved by a correction of manually specified concepts. The analysis shows that the corrections make intuitive sense, most of them are shifts of well-known and easily recognizable countries such as Russia or Iceland to the block of well-known countries (union of Central, Western and Southern Europe).
Table 4.3: Comparison of the manual, automatically corrected, and automatic concepts. Quality of concepts is expressed as RMSE improvement of the concept model with these concepts over the basic model.

<table>
<thead>
<tr>
<th>Concept Type</th>
<th>Number of Concepts</th>
<th>RMSE Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Geography</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manual – type</td>
<td>9</td>
<td>0.00105</td>
</tr>
<tr>
<td>corrected – type</td>
<td>9</td>
<td>0.00070</td>
</tr>
<tr>
<td>manual – location</td>
<td>23</td>
<td>0.00124</td>
</tr>
<tr>
<td>corrected – location</td>
<td>23</td>
<td>0.00106</td>
</tr>
<tr>
<td>manual – combination</td>
<td>54</td>
<td><strong>0.00189</strong></td>
</tr>
<tr>
<td>corrected – combination</td>
<td>54</td>
<td>0.00184</td>
</tr>
<tr>
<td>automatic</td>
<td>5</td>
<td>−0.00029</td>
</tr>
<tr>
<td>automatic</td>
<td>20</td>
<td>0.00005</td>
</tr>
<tr>
<td>automatic</td>
<td>50</td>
<td>0.00009</td>
</tr>
<tr>
<td><strong>Anatomy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manual – system</td>
<td>15</td>
<td>0.00019</td>
</tr>
<tr>
<td>manual – location</td>
<td>10</td>
<td>−0.00004</td>
</tr>
<tr>
<td><strong>manual – combination</strong></td>
<td>90</td>
<td><strong>0.00049</strong></td>
</tr>
<tr>
<td><strong>Europe</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manual</td>
<td>3</td>
<td>0.00003</td>
</tr>
<tr>
<td>corrected</td>
<td>3</td>
<td><strong>0.00012</strong></td>
</tr>
<tr>
<td>manual</td>
<td>6</td>
<td>−0.00016</td>
</tr>
<tr>
<td>corrected</td>
<td>6</td>
<td>0.00003</td>
</tr>
<tr>
<td>automatic</td>
<td>2</td>
<td>0.00007</td>
</tr>
<tr>
<td>automatic</td>
<td>3</td>
<td>0.00005</td>
</tr>
<tr>
<td>automatic</td>
<td>5</td>
<td>−0.00018</td>
</tr>
</tbody>
</table>
5 Response Times

Introduction

Educational data mining research mostly focuses on the correctness of answers which reflects the character of the majority of ITS where correctness is the main measure of success. However, response time (time required to answer an item by a learner) holds information about the solving process and can be a source of extra information. This information can provide an alternative view of the domain model, improve learner models or offer new insight into educational data. Furthermore, this additional information can allow some techniques to work when only a small dataset is available and only correctness information is not sufficient to provide satisfying results.

Response times intuitively have multiplicative nature, i.e., they are dependent on the length of the total time. A faster learner has response times 0.9 times shorter but not 5 seconds shorter across various items with different median response times. This multiplicative nature is also supported by several studies [129, 130, 61] which suggest that response times are log-normal distributed. Therefore, it seems reasonable to work primarily with the logarithm or response time in most cases. In the following section, we denote response time by \( t_{il} \) and its logarithm \( \log t_{il} = \tau_{il} \).

Outline

Most of our publications are focused on time information entirely [16, 83, 82, 86] or include an analysis of the impact of response time use [108, 110, 109, 111, 97]. In this chapter, we summarize our approach to response time use in learning analysis and our experience with its benefits. Section 5.2 gives a general overview of response time properties and shows its usefulness for learner modeling and learning analysis. Section 5.3 introduces our approach to combine the correctness of answer and response time into a single performance measure. The rest of the chapter is devoted to a description of the benefits of this new measure for learning modeling (Section 5.4) or for item similarity measures and mastery learning (Section 5.5).
5. Response Times

5.1 Related Work

Most of the related work is focused on the modeling of response time or the integration of response time into traditional learner models. However, these approaches have their conceptual issues and the author of Conceptual Issues in Response-Time Modeling [130] proposes a hierarchical modeling framework which works with separate models for response and response time but their parameters are modeled in the second level of modeling. In contrast, our approach is to combine both the response and response time into a single measure and use only one model.

5.1.1 Modeling of Response Time

Firstly, we give an overview of the modeling approaches of response time as a standalone variable. The modeling of response time is analogous to learner modeling as described in Chapter 4. Both types of modeling have a similar principle, the difference is that the output for response time modeling is the expected response time and the output of learner modeling is the probability of a correct answer.

A prime example of response time modeling is Rasch’s treatment of oral reading tests [102]. Rash described a model for predicting the time necessary to finish a reading test. He suggested a decomposition of the reader speed parameter $\lambda_{il}$ into the difficulty of the text $d_i$ and the speed (ability) of the reader $\theta_l$ in such a way that $\lambda_{il} = \theta _l d_i$. The total time then can be expressed as $t_{il} = d_i + \theta _l + \epsilon_i$. The log-form of this model is

$$\tau_{il} = d_i - \theta_l + \epsilon_i,$$

where $\epsilon_i$ is a normally distributed random residual. Note that this model is very similar to the 1PL IRT model only the logistic function is replaced by an exponential function.

The next model extends this analogy and was developed for use in the system called Tutor [62, 61]. In contrast to the majority of ITS, this one is focused on items for which the response time is the only measure of success. Typical examples of such items are logical puzzles (e.g., Sudoku, Sokoban) where only correct solutions are allowed but what is important is the time needed to find a solution. There are also
similar but educationally focused problems that focus on the practice of mathematics and informatics.

This model is analogous to the 3PL IRT model and assumes a normal distribution of logarithms of response times in the form:

\[ \tau_{il} \sim \mathcal{N}(d_i - a_i \theta_l, c_i). \]

The difficulty parameter \( d_i \) is interpreted as the expected log-time of an average learner. The discrimination parameter \( a_i \) has the same interpretation as in the IRT model. The variance \( c_i \) expresses the extent of randomness of the item.

The model offers various extensions, e.g., a multidimensional skill which is constructed similarly to the multidimensional extension of the IRT model. Another example can be an introduction of the learning rate parameter which deals with the fact that the skill of a learner is not constant, but it changes over time. The learning rate parameter \( l_i \) deals with this change by taking into account the time spent in the system (or the number of solved items or other measures of work) \( \Delta t \) and the skill is adjusted by \( \theta = \theta_0 + l_i \Delta t \) where \( \theta_0 \) is the initial skill of the learner.

In case that the IRT model is used to model the correctness of answers, it is possible to use its parameter to improve Rash’s basic response time model. Essentially, these models add a new term to the log-form of this model. Thissen’s model [125] adds the term

\[ -\rho(a_i \theta_l - b_i) \]

which incorporates parameters from the 3PL model \((a_i, \theta_l \text{ and } b_i)\). The real parameter \( \rho \) expresses the degree of influence of a new term on the response time. A similar approach is used in Gaviria’s model [47], which distinguishes between the correct and the incorrect answer in the model, or a model described by Ferrando and Lorenzo-Seva [44], which replaces \( a_i \theta_l - b_i \) by \( \sqrt{a_i^2 \theta_l^2 - b_i^2}^2 \).

### 5.1.2 Response Time in Learner Modeling

In this section, we focus on the research about the relationship between the response time and the correctness of answers. Previous research [136, 75] suggests that a shorter response time is linked with
a higher success rate. At first sight, this can seem counter-intuitive. There is accuracy-speed trade-off, i.e., we expect a better result with a higher response time. However, this assumption only holds when looking at one learner. When we aggregate observations of many learners, we can see that skilled learners who are confident with their answers are typically faster than learners who are not sure and have to think about their answer more.

This link between correctness and response time was successfully used to directly improve the adaptivity of a system [78] (adjusting the spacing algorithm). Another approach to the utilization of response time is to use learner models to improve predictions of correctness. Due to our focus on learner modeling based on the logistic function, we discuss mainly an extension of the IRT model, but there are also applications of response time with the BKT model [136].

Response Time in IRT

Models which combine response times and classical IRT assume a trade-off of accuracy (can be understood as skill) and the speed of the learner. In some contexts, it is reasonable to take a longer response time (lower speed) as a compensation for a missing skill. To some extent, speed is a choice of the learner, thus the skill can be expressed as a function of the speed. It is assumed that this function is decreasing but, generally, this function has an unknown shape which is dependent on the skill, the item, and the learner.

One of the first IRT models working with the time information was introduced by Roskam [114, 115]. This model is based on the 1PL model and assumes a linear trade-off of the skill and the logarithm of response time:

\[
P(\text{correct}, \theta_l) = f(\theta_l + \tau_{il} - d_i) = \frac{1}{1 + e^{-(\theta_l + \tau_{il} - d_i)}}.
\]

This model increases the probability of success with response time on a given item. In other words, an answer with a long response time testifies of a lower skill of a learner than an answer with a short response time.
The model presented in [132] replaces the time information $\tau_{il}$ by a speed parameter $\tilde{\theta}_l$:

$$P(\text{correct}, \theta_l) = f(\theta_l + \tilde{\theta}_l - d_i)$$

The speed parameter is computed from response times and gives a better chance of success for faster learners. In contrast to the previous model, which assumes variability of response times, this one is more suitable for situations where time for answering an item is limited.

The next model described in [135] extends the 3PL model and besides the speed parameter $\tilde{\theta}_l$ (in this context taken as the slowness parameter $\sigma_l = 1/\tilde{\theta}_l$) and response time information also takes the slowness (or labor intensity) of item $\delta_i$ into account:

$$P(\text{correct}, \theta_l) = c_i + (1 - c_i)f(a_i(\theta_l - \sigma_l\delta_i - t_{ij} - d_i)).$$

This model shows the structural difference with respect to the accuracy-speed trade-off in Roskom’s model, where the probability of success goes to 1 with increasing time whereas this model converges to a regular 3PL model.

**‘High Speed High Stakes’ Scoring Rule**

Maths Garden ITS [68] imposes the speed-accuracy trade-off setting on the learner. The system uses a high speed high stakes (HSHS) scoring rule [77] which decreases the gain or loss of a score with the running time. The scoring rule is also projected to the Elo model [68] which is used in the Maths Garden system. The main idea is to transform the binary response $r_{li} \in \{0, 1\}$ of a learner $l$ on item $i$ to a score $s_{li} \in [-1, 1]$ in such a way that the score drops to zero with time increasing to a limit $d_i$ (score for response time $t_{li} > d_i$ is always zero):

$$s_{li} = (2r_{li} - 1)\frac{d_i - t_{li}}{d_i}.$$

This method is closely related to our approach to combine response and response time described in Section 5.3. The main difference is that we never use response time information for wrong answers and thus our equivalent of the score lies in $[0, 1]$. 

101
5. Response Times

Figure 5.1: The basic characteristics of learners divided into 10 groups of the same size based on their median response time. Left: the average number of answers in the system; Right: the time in seconds spent practicing computed as the sum of response times.

5.2 Descriptive Statistics

First of all, we are interested in whether the response time carries interesting information about learners at all. Therefore, we performed the following simple analysis. We computed the median response time for each learner with at least 10 answers, divided learners based on these median times to 10 fastness bins and computed basic statistics for these groups. Figure 5.1 shows the results for 3 systems.

For the geography and anatomy datasets, we found that slower learners have a smaller number of questions in the system. In fact, the overall time in the system is nearly the same for learners with different speeds, i.e., slower learners just answer a smaller number of questions during this time. Moreover, in the geography dataset, faster learners have a higher prior skill (success rate 0.81 for fast learners compared to 0.72 for slow ones) and are more likely to return to the system to do more practice (18% return rate for fast learners compared to 10% for the slow ones). In the feedback on question difficulty, slower learners report a more difficult impression, on a scale of 1 to 3, meaning: 1 easy, 2 appropriate and 3 difficult, the average rating of fast learners is 1.7 compared to 1.9 for slow learners. A possible application of these
5. Response Times

results is an incorporation of learners’ speed into the algorithm for an adaptive selection of questions (e.g., by selecting easier questions for slower learners).

The results for the matmat system are different and suggest that the number of answers is almost independent of the learners’ speed and thus the time spent in the system is longer for slower learners. This is probably due to a higher system relevance for younger learners, who tend to be less skilled and thus slower but the system is more relevant for them. Overall, these results suggest that the response time carries useful information about learners.

Prediction Power

Most of all, we are interested in whether the response time can be used to determine learner’s skills and to predict the correctness of their future answers. Therefore, we study the relationship between response time and correctness of answer.

Response times clearly depend on the type of question and specific item. For example, our results for the geography dataset show that response times are higher for cities and rivers than for countries and regions (states are larger than cities on the used interactive map and therefore it is easier to click on them). Response times are also higher on average for countries in Asia than in South America (there is a larger number of countries on the map of Asia).

For the below-presented analysis, we use percentiles of response times over individual items – these are not influenced by skew and provide normalization across different items.

Figure 5.2 shows the relationship between response times and the correctness of answers for the Outline maps system. The relationship between response time and correctness of the current (purple line) answer is non-monotonic – very fast responses combine solid knowledge and pure guessing, long responses mostly indicate weak knowledge. The highest chance of correct answers is for response times between 10th and 20th percentile, i.e., answers that are fast, but not extremely fast.

We get a more straightforward relationship when we analyze the correctness of the next answer about the same item (blue line) based on both the correctness and response time for the current answer. If the current answer is correct (red line) then the probability of the
next correct answer is linearly dependent on the response time – it goes from 95% for swift answers to nearly 80% for slow answers. If the current answer is incorrect (green line) then the dependence on response time is weaker, but there is still an (approximately linear) trend, but in this case in the opposite direction. When the current answer is incorrect, a longer response time actually means a higher chance that the next answer will be correct.

A limitation of the current analysis is that we do not take into account the types of questions (the number of available choices and the related guess factor) or the adaptive behavior of the system (the system asks easier questions when knowledge is estimated to be low). However, we do not expect these factors to significantly influence the reported results, which quite clearly show that response times are useful for modeling knowledge and that it is important to analyze response times separately for correct and incorrect answers.

This result with data from the Outline maps system clearly shows promising results even though the focus of the system is on the memorization of facts, and we do not expect response time to play an important role. The same analysis for the MatMat system is unfor-
5. Response Times

Fortunately not entirely possible because there are not enough learners with multiple answers to one item in the dataset. Thus we evaluated the success rate for the current answer and globally next answer, i.e., independently on the answered item. However, the consecutive items in MatMat practice are more closely related than in Outline maps. Results (see Figure 5.3) show that the relationship between speed and the success rate is less pronounced than for geography. This suggests that the response time is not so beneficial for the prediction of correctness here. However, this does not mean that response time does not carry important information about the learner ability in a different direction.

Figure 5.3: Response times and probability that the (next) answer is correct. Data comes from MatMat system.
5. Response Times

5.3 Combining Correctness and Response Time

Most learner modeling approaches and educational data mining techniques consider only the correctness of answers. Response times, however, can provide useful information about learner knowledge, particularly for the domain of basic mathematics. In this section, we use a novel approach to combine binary information about correctness and response time information into a single performance measure. This measure can be then used in proven techniques and models, which work typically with only a binary value, only with small adjustments. An alternative to this approach is to treat both pieces of information separately, but it is usually at the cost of this portability.

We define the possible measures $r \in [0, 1]$ in such a way that 0 means failure of the learner and 1 complete success. Thus, these measures are compatible with the standard binary measure (correctness) but add the possibility to express more detailed information about success or failure. We give several types of this measure. A wrong answer is understood as a failure and thus we always ignore response times (i.e., $r$ is constantly equal to 0) for this type of answers. For correct answers, we use one of the following functions (see also Figure 5.4):

- **noTime** – no use of time; $r = 1$, $r \in \{0, 1\}$.

- **thresholdTime** – response is classified as fast or slow based on the threshold $\lambda$ seconds; $r = 1$ for fast responses, $r = 0.5$ for slow responses, $r \in \{0, 0.5, 1\}$.

- **expTime** – similar to the previous one; for fast responses $r = 1$, for slow responses $r$ decreases exponentially $r = x^{t/\lambda - 1}$. The parameter $x < 1$ determined how fast $r$ drops with response time and usually we set $x = 0.9$.

- **linearTime** – response is linearly decreasing with increasing time (until it reaches 0): $r = max(0, 1 - t/2\lambda)$.

The threshold $\lambda$, used in all nontrivial measures, determines which response times are considered fast enough. This can be set manually but we mostly use the data drive approach, and we set $\lambda$ to the median response time. This can be done separately per item or, if items are similarly time intensive, globally for the whole system.
5. Response Times

The described measures have other parameters (e.g., $x$ in $expTime$) and variants (e.g., $thresholdTime$ with more thresholds and more possible values). We are not able to easily evaluate which approach or what parameters are optimal and this optimum can probably differ depending on the purpose and use of response time. Instead, we focus on proving validity and usefulness of response time utilization as such. For this purpose, we order measures from the least to the most pronounced use of response time: $noTime$, $thresholdTime$, $expTime$ and $linearTime$.

Note that a similar approach of combining more types of information to a single measure of success can be used with other types of additional data, such as the number of used hints, number of trials, type of wrong answers [96] or other data provided by a system.

5.4 Learner Modeling

In this section, we study how the use of the combined performance measure impacts learner models described in Chapter 4.

5.4.1 Evaluation Issues

To analyze the usefulness of response time utilization in learner modeling, we analyze the approaches described in 5.3 on the data from the MatMat system. To get more robust results, we use several learner models based on different domain models described in Chapter 4. For
5. Response Times

Figure 5.5: Model comparison according to RMSE and AUC with and without time information. Error bars show 95% confidence intervals. Note that for the RMSE metric, lower values mean better performance, whereas for the AUC metric, higher values are better. Also, note that the RMSE values of different time uses are not straightforwardly comparable because models are trained to predict different absolute values.

a comparison of the predictive accuracy of models, we use a rather standard evaluation setting: repeated random cross-validation (20 runs) with learner stratified train/test set division (70%/30%)

The evaluation of learner models which consider timing information is difficult. Standard performance metrics [94] consider only binary information about correctness, but the point of models with timing information is to also distinguish between slow and fast responses. We report three selected metrics to give basic insight into the model behavior. First, we use the standard RMSE (i.e., the performance metric ignores the timing information). Secondly, we use a “ternary” version of RMSE where observations are labeled as 0 (wrong), 0.5 (correct, slow), and 1 (correct, fast). Thirdly, we use the Area Under the ROC Curve (AUC) metric, which ignores the timing information but considers the prediction only in a relative manner.

The comparison of various models with a different type of time use according to these metrics is shown in Figure 5.5. With respect to the complexity of the used domain model, the results are similar to the results presented in Section 4.4.3. Also, note that the modeling of the timing information is rather orthogonal to domain modeling. With respect to different variants of modeling response times, we see
large differences between models, but these differences are not easy to interpret due to the impact of the used metric.

The results for the two variants of RMSE are not very surprising since RMSE takes into account absolute values of predictions and different models are trained to predict different values, e.g., models with `linearTime` have a much higher RMSE since they are trained to give much smaller predictions (with a different meaning). The best models with respect to the reported RMSE metrics are those that match the performance metric used for evaluation. The interesting part of results is the comparison with respect to AUC. Although the evaluation metric does not take response time into account, the best results are achieved using `linearTime` model variants. This modeling approach improves the relative order of the predictions. However, a comparison with respect to AUC assumes that slower learners are relatively more likely to make a mistake than faster learners with the same overall correctness. It is questionable to what extent this assumption is met, e.g., some high-skilled learners can be overall slower due to their different method of work or different use of the system’s interface (use of the mouse instead of the keyboard, etc.).

We can construct different metrics taking into account response time, but even this simple example shows that the selection of a metric has a great impact on results. This is mainly because each metric gives importance to a different part of the data and defines the criterion for a good model in a different way. However, we are not able to say which criterion and thus which metric is the right one for our system. Therefore, we are not able to use this straightforward approach to compare and evaluate a different type of time utilization and we search for other evaluation methods in the following sections.

### 5.4.2 Impact on Estimated Parameters

To get insight into the differences between time uses, we analyze the correlations between parameter values, particularly the item difficulty parameters which have clear interpretation and a direct impact on the adaptive system behavior (e.g., adaptive selection of items or provided feedback for learners and teachers).

Figure 5.6 shows the correlations between item difficulties for different models. A darker color means a higher correlation, i.e., more simi-
5. Response Times

Figure 5.6: Spearman correlation coefficient for estimated item difficulty parameters of different models.

Similarity in the difficulty estimates (a less important difference between models). Unsurprisingly, there is a large gap between the baseline model and other more sophisticated models. The impact of the learner modeling approach is non-trivial, but not pronounced. Different utilization of time, however, brings considerably different parameters. The degree of change is proportional to the intensity of time utilization, \( \text{linearTime} \) extension is the most different. Also, note that the figure contains a repetitive \( 4 \times 4 \) pattern corresponding to a different time usage. This means that domain modeling and time modeling are almost independent modeling aspects and provide change (and possible improvement) in different directions.

To provide better intuition beyond the summary of the evaluation metrics and correlations, we provide a specific illustration of the model impact on the parameters of simple addition and subtraction items. Figure 5.7 presents a comparison of the estimated parameters for a model with and without timing information. The estimated parameters are only weakly correlated and there are significant differences (e.g., \( 1 - 1 \) and \( 8 - 5 \) have the same difficulty according to the model without response times, but quite a different difficulty according to the model with time). Particularly, note the highlighted subtraction
5. Response Times

examples of the type $X - X$ which the model with timing information systematically rates as easier than the model without time.

Figure 5.7: Comparison of estimated difficulties of the selected items by the basic model with and without time usage. Note that absolute values of difficulties are not entirely comparable because models are trained to predict different values.

5.4.3 Parameter Stability

Before we use a model in an adaptive educational system, we want to be sure that its parameters are reasonably stable. For example in Figure 5.7, we can see that the estimated parameters are probably not completely stable yet (one would intuitively expect for example that the items $X - X$ would either have a very similar difficulty or be ordered). How to objectively judge parameter stability? How quickly do parameter values stabilize? How much do different models differ in their speed of convergence? Such questions do not get much attention in learner modeling. A recent exception is a proposal for a multifaceted evaluation of learner models [57] where authors include parameter stability as a criterion for model evaluation, but they discuss only a specific model (BKT) and do not focus on the dynamics of parameter stability.

To evaluate these questions, we performed the following experiment. We took two data samples $D_1, D_2$ of size $K$ (learner stratified,
5. **Response Times**

without intersection). We considered a particular model type \( M \) and trained an instance \( M_1 \) using the data set \( D_1 \) and an instance \( M_2 \) using the data set \( D_2 \). Then we evaluated the correlation of parameters of \( M_1 \) and \( M_2 \) (we used only item difficulty parameters since the data sets are learner stratified).

Figure 5.8 shows an increase in parameter stability with the number of answers used for training the model. The figure compares different domain modeling approaches and different time uses. The left part of the figure shows that the hierarchical model is slightly more stable than simpler models as it can carry information across items better. The right part of the figure shows a high increase in stability of models which utilize response times. Figure 5.7 provides a specific illustration, note that the group of similar items of the form \( X - X \) have very similar difficulty according to the model utilizing response time, but widely different difficulties for the model without response times. This increase in stability (resp. faster convergence) is probably mainly due to the use of more bits of information per each answer. The increase in stability is also proportional to the intensity of time utilization.

![Figure 5.8: Parameter stability: correlation between item parameter values for models fitted on two samples of a given size. Left: Comparison of different learner models. Right: Comparison of different response time models. Confidence intervals are mostly too small to be visible.](image-url)
5.5 Other Benefits of Response Time Usage

5.5.1 Item Similarity Measures

The correctness of answers is the basic source of information about item similarities described in Section 3.2, but the similarity of items can also be computed based on measures defined in 5.3. In this analysis, we use linearTime due to its simplicity and high influence of response time information. The measure obtained as a combination of correctness and response time is a continuous variable, therefore it limits possible similarity measures, and we used the Pearson measure.

Different Point of View

Figure 5.9 shows scatter plots of item pairs with position determined by their similarity with and without the use of response time. The correlation of similarity measures varies depending on the dataset but for all matmat datasets, the correlation is low enough to conclude that the response time brings a significant change to the similarities of the items and therefore the incorporation of response time information to the similarity measure can change the meaning of similarity.

Figure 5.10 gives such an example and shows the projection of items from matmat-numbers focused on practicing number sense. Similar items according to measures using only the correctness of answers tend to be items with the same graphical representation in the system. On the other hand, similar items according to measures also using response time are usually items containing numbers of a similar value.

Speed of Convergence

We also used this method on data sets from Math Garden (specifically mathgarden-addition), which are much larger. In this case, the use of response times has only a small impact on the computed item similarities (correlations between 0.85 and 0.95). However, the use of response times influences how quickly the computation converges, i.e., how much data we need. To explore this, we considered as the ground truth the average of computed similarity matrices with and without response times for the whole data set. Then we used smaller samples of the data set to compute item similarities and checked the
5. Response Times

Figure 5.9: Comparison of double Pearson similarity values with and without the use of response time (linearTime approach was used) for math datasets.

agreement with this ground truth. Figure 5.11 shows the difference between the speed of convergence of the measure with and without response time utilization. Results show that the measure which uses additional information from response time converges to ground truth much faster. This result suggests that the use of response time can improve clustering or visualizations when only a small number of answers is available.

5.5.2 Mastery Detection

Mastery learning is an instructional strategy that requires learners to master a topic before moving to more advanced topics. A key aspect of mastery learning is a mastery criterion – a rule that determines
5. Response Times

Figure 5.10: Projection of items practicing number sense from the Mat-Mat system. Left: Measure based only on correctness. Right: Measure using response time. Opacity corresponds to the number value of the item and color corresponds to the graphical representation of the task.

whether a learner has achieved mastery. We studied evaluation methods and compared various mastery criteria in our work [97] about mastery learning. One of the studied aspects of mastery detection is the impact of response time.

In the following experiment, we use real data from the MatMat system and explore the relative importance of the choice of a model and the choice of usage of learners’ response times. In the case of basic arithmetic, it makes sense to include fluency (learners’ speed) as a factor in the mastery decision. Does it matter whether we include response times? If so, how much?

For the detection of mastery, a learner model can be used. Mastery is declared when the estimated skill of the learner exceeds a given threshold. For the choice of a skill estimation model we consider the following two variants:
5. Response Times

Correlation with ground truth
Sample size
with
without response times

Figure 5.11: The speed of convergence to ground truth for measures with and without response time on Math Garden addition data set.

- The basic exponential moving average (EMA) method with exponent $\alpha = 0.8$. This method computes an average of the previous responses in such a way that recent ones have a larger impact on the final estimation. The EMA method is described in more detail in [97].

- A hierarchical model (denoted as $M$) described in detail in 4.3.2.

The basic difference between these two approaches is that the model takes into account the difficulty of items, whereas the EMA approach completely ignores item information. Therefore, the model can deal with the adaptively collected data better.

For the choice of input data we also consider two variants:

- $noTime$ — only the basic binary correctness data.

- $linearTime$ — combination of correctness and response times, denoted as $+T$.

We compare four models obtained as combinations along these two dimensions: EMA, EMA+$T$, M, M+$T$. We evaluate agreement between them to see which model aspects make a larger difference. To analyze mastery decision, it is necessary to choose mastery thresholds.
However, the studied methods differ in the scales of their output values, e.g., EMA + T gives smaller values than EMA. It is therefore not easy to choose thresholds for a fair comparison. To avoid biasing the results by a choice of specific thresholds, we directly compare orderings of learners by different methods. For each learner, we compute the final skill estimate, and we evaluate agreement of methods by the Spearman correlation coefficient over these values.

Figure 5.12 shows the correlations of the four studied methods for four knowledge components from the MatMat system. We see that the correlation between EMA and the model approach is typically higher than the correlation between approaches with and without the use of response time. Particularly the variants with timing information (EMA+T and M+T) are highly correlated. From this analysis, we cannot say which approach is better, but we see that the impact of using response times is larger than the impact of using a learner model.

Figure 5.12: Spearman correlation between different learner skill estimation methods for different knowledge components (Matmat data).
6 Conclusion

In this thesis, we studied various techniques with multiple goals and applications. In this chapter, we summarize the main results of our work, give general impulses for related research and outline topics for future work.

6.1 Specific Results

Development of Adaptive Systems

One of the practical outputs of our work is the MatMat system, which was designed and developed in symbiosis with our research. The system focuses on the domain of basic arithmetic, for which response time has a major role in the estimation of learner proficiency, and it was a valuable source of data for our analysis. On the other hand, the system benefits from our research results and implements theoretically proven models.

Domain Modeling

The first step in our approach to domain modeling is the computation of item similarities. An important part of this step is the choice of a suitable similarity measure. Our results provide some guidelines for this choice. The Pearson, Yule, and Cohen measures lead to significantly better results than other compared metrics. It is also beneficial to use the second step of item similarity. The specific choice between the mentioned measures in the first step and the choice of measure in the second step does not seem to make a fundamental difference. However, the Pearson correlation coefficient is a good default choice, since it provides quite robust results and is applicable in several settings and both steps.

We studied how to use these similarities in domain modeling. We focused mainly on a classification (assigning each item to a single concept) as an alternative to the Q-matrix approach. For this purpose, we used clustering algorithms and compared the most common approaches (k-means and hierarchical clustering) and one rather novel
6. Conclusion

approach in the educational context (spectral clustering). The results suggest that all three approaches are applicable for detection of concepts from both the correctness of answers and solving times. For correctness results, hierarchical clustering performs slightly worse; and for solving times, spectral clustering outperforms the other two methods. As the first choice, we recommend using k-mean due to its simple implementation, easy use without the necessity to set many parameters and robust results. For a deeper analysis, we recommend spectral clustering for its ability to provide additional insight into data.

We also examined the possibility to use item similarities to find mistakes in concepts provided by a human expert. We proposed the use of supervised learning methods (we used kNN and logistic regression) to address this task and study the conditions (error rate of a human expert) of bringing advantage to this approach over purely expert or data approach. The results showed that this zone of error rate grows with the number of skills and it is sufficiently large to deserve further attention.

We used these automatic or semi-automatic approaches to create concepts and used these concepts in a learner model. The result showed no or only a small improvement of the accuracy of the model. However, these methods often have other outputs, which can be used to get insight into a domain. Hierarchical clustering provides a whole hierarchy of items; spectral clustering offers alternative visualization of items; found mistakes in the expert’s concept can point not only at mistakes but also at outlier items; etc. To get insight into the domain, we also study visualization methods. Our analysis suggests that, a relatively new approach, t-SNE is a visualization technique with easily interpretable outputs and compares better with classical methods like PCA and MDS.

Learning Modeling

The second area is learner modeling in which we focused on online models that are easily usable in an environment of a real educational system. We study primarily Elo based learner models whose main idea is to update estimated skills and difficulties proportionally to the unexpectedness of the learner’s answer. For the basic Elo model for
6. Conclusion

Prior knowledge estimation, we introduce two alternatives: the concept and the network model. The results show that if we are concerned only with the accuracy of predictions, the basic model is a reasonable choice. More complex models do improve predictions in a statistically significant way, but the improvement is relatively small.

The improvement in predictions by the concept or networked models may be more pronounced in less homogeneous domains or with less homogeneous populations. Nevertheless, if the main aim of a learner model is the prediction of future answers (e.g., applied for the selection of questions), then the basic model seems to be sufficient. Its performance is good and it is effortless to use. Thus, we believe that it should be used more often both in implementations of educational software and in evaluations of learner models. The more complex models may still be useful since improved accuracy is not the only purpose of learner models. We used them in an evaluation of the quality of the domain model or they give more complex information about learners’ skills.

We also introduced the hierarchical model which deals with the estimation of prior and current knowledge simultaneously. This model was specially designed to work with a more complex domain model (represented as a hierarchy of concepts), which was required by the domain of basic mathematics. This model allows propagating information from an answer to the item in one concept to other concepts. Despite a relatively high number of tracked skills and a small number of answers per learner in the dataset, the model performs better compared to the PFAE model. We deployed the hierarchical model on the MatMat system. There, it successfully deals with the main challenge: the tracking of very specific skills while maintaining a fast estimation of the skills of new learners.

Response Time

The last studied area is the utilization of response times. We showed that response time itself carries interesting information about learners and that there is a correlation between the correctness of answers and response time and it can be used in predictive models. We also introduced a simple approach to combine the correctness of answer and response time to a single measure, which can be used in learning anal-
6. Conclusion

ysis instead of the correctness of answer without substantial changes in algorithms. Even though the results based on a combination of the correctness of answer and response time are difficult to compare to the classical approach, we give several examples of the benefits of the use of response time.

The use of response time allows us to redefine success from correct answers to correct and fast answers. Our results show that this new definition can bring an alternative point of view on the similarity of items, the difficulty of items or mastery. Our experiments also suggest that extra information provided by the use of response time increases the stability of item similarities or estimated parameters (e.g., difficulties) of learner models, which is especially useful when we have available datasets with only a limited size.

6.2 Lessons Learned

The previous section gives an overview of specific results presented in this work, but the thesis also analyzes more general research questions in the area of educational data mining.

6.2.1 Modeling Techniques

Modeling for Automatization

We introduced novel learner models and proved their benefits. However, the main message in the area of learning modeling is that Elo based models are a good alternative to commonly used models such as BKT. As previous research showed, their performance is comparable to other approaches and, as described in this work, they are relatively simple and thus can be easily implemented in educational systems, but they also offer various more complex extensions and modification and thus are interesting for research.

Our analysis of response times showed that this type of information is easily usable and it is beneficial in various ways. It can improve the results of automatic methods or allow these methods to work with different data. Therefore, we believe that response time and other alternative sources of information about learners should be included in future models and learning analysis more often.
Modeling for Leveraging Human Intelligence

Some of the techniques described in this work aim at an automatization of the development and maintenance of adaptive systems. However, many techniques are in line with Baker’s argument [8] for focusing on the use of learning analytics for leveraging human intelligence instead of its use for automatic intelligent methods. Therefore, the output of the introduced techniques is not often straightforwardly usable in an adaptive system, but they are a basis for work and decisions of developers and experts.

Throughout the work, we pay attention to making outputs which are interpretable, intelligible and which allow humans to understand data better. From this perspective, the most important area is domain modeling, where we described a general two step approach: a computation of item similarity and a follow-up usage of these similarities for the visualization of a domain, automatic domain construction, etc. We studied these steps separately and gave a strong foundation for the choice of a similarity measure in the first step so researchers can focus on the second step, which offers much more variety and possibilities. We also gave examples of such second step techniques whose goal is to make the work of experts easier: visualize the relationships of items, find mistakes in manual labeling, suggest concepts, etc.

6.2.2 Evaluation Methodology

An important part of this work is also the evaluation process of old and novel techniques. We used several evaluation methods, some of them standard but some of them rather less known or used. Many of them are not bound to an evaluation technique but can also be used in other scenarios. Since evaluation has a key role in the scientific process, we give a summary of the main evaluation approaches we used in this work, which can serve as an inspiration for appraisal and understanding of new techniques.

Performance Evaluation

Probably the most reliable and widely used evaluation method is the comparison of the output of the evaluated technique to the expected result (ground truth). Even this straightforward evaluation method
6. Conclusion

has its pitfalls. First of all, as described in [94], the choice of metrics used in the comparison is important and influences the results and conclusions based on them. But other factors are also important. For example, circumstances data collection is a factor which has to be considered by researchers. This is of particular importance in the context of adaptive systems where items are not presented in a random order but an order based on the learner model and other conditions. Other factors which can influence results are the division to a train and a test set or the order in which we process data. More about these and other aspects of this type of evaluation is described in our work [97].

We used this basic approach whenever we had the ground truth available. Specifically, we used this method in the evaluation of learner models in Section 4.4. As a comparison metric, we used RMSE which, as we believe, is the most meaningful one in the context of learner modeling. However, we also reported other metrics like AUC or LL. A noteworthy metric is also the Brier score which is basically only a decomposition of RMSE but can give (as shown in Section 4.4.1) useful insight into the behavior of models. The comparison with the ground truth was also used in the evaluation of similarity measures and clustering methods in Section 3.3.3. Here, an adjusted Rand index was used to compare manual clusters and clusters provided by automatic techniques.

However, the ground truth is not often available, or we are not able to say what exactly the optimal result should be. This issue can be seen in the evaluation of learner models with response times in Section 5.4.1. Therefore, we are forced to look for other methods which can give us insight into the behavior and usefulness of the used techniques.

Simulated Data

The first method of dealing with a missing ground truth in available real world datasets is to use simulated data. For this type of data, we know exactly how they were generated and what the underlying parameters are. We used this technique to evaluate similarities in Section 3.2.3, clustering methods in Section 3.3.3 or compare visualization methods in Section 3.5.2. Of course, this evaluation approach has its limitations. Simulated data are only a simplification of reality, and datasets from real systems can have different characteristics, thus re-
6. Conclusion

Results with simulated data do not necessarily correspond to the results with real world data. Moreover, there are many ad hoc choices about the parameters of the simulation that have to be made. One example is the ratio of noise in data which can make the task too easy or too difficult. However, this method can give a good initial insight into novel methods and can serve as a test of a new concept. Also, this approach allows simulating scenarios which would not be possible in real systems.

Parameter Stability

Another evaluation method which we used repeatedly is the computation of parameter correlation of two models or techniques. We use the correlation of similarity values in Section 3.2.3, the correlation of estimated difficulties and skills in Sections 4.4.4 and 5.4.2 or in depicting influence of response time on mastery detection in Section 5.5.2. It is not possible to use this approach to conclude that one method or model is better than another, but that the new method has significantly different parameters or results, and therefore it is worthy of further exploration. The new method can have better results or can give a different point of view on the data as the examples in Section 5.5 show.

The last evaluation method is an estimation of the stability of the used model or technique. As above, the correlation of parameters or results is computed, but in this case, we have one technique used with two different samples of a dataset. This approach allows us to judge whether our dataset is sufficient to get stable results. If we get a high correlation of parameters or result for two sample datasets, we can conclude that we have a stable result. A low correlation means that more data is needed to get meaningful results. We used this method for the evaluation of similarity measures in Sections 3.2.4 and 5.5. Figure 3.7 gives a good example of this evaluation method and shows the differences between datasets of various sizes and their sufficiency.

We cannot expect easy and definitive answers in learning analytics. Therefore, we have to try to understand techniques in depth to be able to set them appropriately and use them correctly and in diverse situations. The evaluation methods described above are not always straightforwardly interpretable but they give us insight into what is
6. Conclusion

happening \textit{inside} our algorithms and allow us to understand their behavior better.

6.3 Future Work

We described a two step approach to analyze a domain which is based on item similarities. In current research on automatic and semi-automatic domain modeling, we limited our focus only to the classification of items. Future research should focus on the question if the item similarity approach can also be used for building more complex domain models such as the Q-matrix approach. It should also focus on the comparison of the item similarity approach and matrix factorization methods. Other interesting directions of research are the study of asymmetrical similarity measures, which can express a different type of relationships such as prerequisite, or the use of the similarity approach for a classification of learners instead of items, which allows us to depict different behavior patterns and potentially different learning needs.

This work studies item similarities in great detail and gives examples of their usage in various directions. But there are other possible applications of item similarities which are beyond the scope of this thesis. The follow-up research should verify whether item similarities can be used to detect outlier items, whose unusual behavior can mean tricky questions or mistakes in items. It should also search for topics with thin item coverage; improve item selection process to avoid repeating similar items in a short time; etc.

In prior knowledge estimation, we found that more advanced models, which are less sensitive to the assumption of homogeneity among learners, perform slightly better than the basic model. A hypothesis worthy of verification is that these models perform even better for the data collected from the systems with more diverse learners and their prior knowledge, e.g., in the context of geography learners from more continents. Also, the hierarchical model gives promising results. As the next step, this model should be studied in more detail, especially in relation to the used domain model. In this work, we used one particular hierarchy of concepts, but future research should answer the following question: What is the best concept hierarchy for basic
6. Conclusion

mathematics? How to extend this hierarchy to cover more of the math curriculum? How to build the concept hierarchy for other domains?

Our analysis of the usage of response time shows promising results, especially in learning analysis. To prove the usefulness of response times in learner modeling with practical impacts on learning, an A/B experiment should be done in a real system. For example, half of the learners would have item selection and feedback based only on the correctness of their answer and for the other half also response time would be used. In this setup, we are interested in the differences in learning gain, motivation, time in the system, etc. Another direction of the future work is the use of the idea of a combination of correctness and response time into one performance measure but with other types of learners’ information such as the number of used hints, the number of trials or other system specific information.
7 Publications

Publications directly related to the thesis

[16] Automatic detection of concepts from problem solving times.
Petr Boroš, Juraj Nižnan, Radek Pelánek and Jiří Řihák
I proposed the use of the spectral clustering method in educational context, implemented and evaluated this method. 30%

Juraj Nižnan, Radek Pelánek and Jiří Řihák
I proposed the use of supervised learning methods for correcting the mistakes of the expert, implemented and evaluated these methods. 35%

Juraj Nižnan, Radek Pelánek and Jiří Řihák
I proposed the use of supervised learning methods for correcting the mistakes of the expert, implemented and evaluated these methods. 35%

[84] Student Models for Prior Knowledge Estimation.
Juraj Nižnan, Radek Pelánek and Jiří Řihák
I designed, implemented and evaluated the concept model and the network model. 40%
7. Publications

[86] **An Analysis of Response Times in Adaptive Practice of Geography Facts.**
Jan Papoušek, Radek Pelánek, Jiří Řihák and Vít Stanislav
I analysed the relationship between the speed of learners and their performance and other characteristics. 25%

[108] **Use of Time Information in Models behind Adaptive System for Building Fluency in Mathematics.**
Jiří Řihák

[110] **What is More Important for Student Modeling: Domain Structure or Response Times?**
Jiří Řihák and Radek Pelánek
I designed, implemented, evaluated and made other analyses of learner models with various aspects (domain model, response time use). 60%

[109] **Choosing a Student Model for a Real World Application.**
Jiří Řihák and Radek Pelánek
I designed, implemented, evaluated and made other analyses of learner models with various aspects (domain model, response time use, wrong answers). 60%

[95] **Elo-based Learner Modeling for Adaptive Practice of Facts.**
Radek Pelánek, Jan Papoušek, Jiří Řihák, Vít Stanislav and Juraj Nižnan
I designed, implemented and evaluated the concept model and the network model. 25%
7. Publications

Jiří Řihák and Radek Pelánek
I introduced the second level of item similarity and I compared and evaluated various similarity measures. 60%

[97] Experimental Analysis of Mastery Learning Criteria.
Pelánek, Radek and Jiří Řihák
I analysed the impact of response time usage on mastery detection. 20%

Other Publications

All these publications are all very closely connected to the general theme of the thesis, but for the sake of the text coherence, the results are not covered within the thesis.

[96] Properties and Applications of Wrong Answers in Online Educational Systems.
Radek Pelánek and Jiří Řihák
I analysed the properties of the categories of wrong answers. 35%

[98] Impact of Data Collection on Interpretation and Evaluation of Student Models.
Radek Pelánek, Jiří Rihák and Jan Papoušek
I analysed the impact of adaptive choice of items and the feedback loop between the learner model and data collection. 20%
Bibliography


BIBLIOGRAPHY


[65] Tanja Käser, Alberto Giovanni Busetto, Gian-Marcio Baschera, Juliane Kohn, Karin Kucian, Michael von Aster, and Markus


[71] Patty Lehner. What is the relationship between fluency and automaticity through systematic teaching with technology (fast math) and improved student computational skills. Virginia Beach City Public Schools, 2008.


BIBLIOGRAPHY


BIBLIOGRAPHY


