User Modeling and User-Adapted Interaction manuscript No.
(will be inserted by the editor)

Beyond Binary Correctness: Classification of Students' Answers in Learning Systems

Radek Pelánek · Tomáš Effenberger

Received: date / Accepted: date

Abstract Adaptive learning systems collect data on student performance and use them to personalize system behavior. Most current personalization techniques focus on the correctness of answers. Although the correctness of answers is the most straightforward source of information about student state, research suggests that additional data are also useful, e.g., response times, hints usage, or specific values of incorrect answers. However, these sources of data are not easy to utilize and are often used in an ad-hoc fashion. We propose to use answer classification as an interface between raw data about student performance and algorithms for adaptive behavior. Specifically, we propose a classification of student answers into six categories: three classes of correct answers and three classes of incorrect answers. The proposed classification is broadly applicable and makes the use of additional interaction data much more feasible. We support the proposal by analysis of extensive data from adaptive learning systems.

Keywords adaptive learning · student modeling · interface · classification · response time

1 Introduction

Adaptive learning systems collect data on student performance and use them to personalize system behavior, e.g., to implement mastery learning, to provide personalized recommendations or feedback to students. The primary source of data that drives this personalization is student interaction with practice items (questions, problems).

The data about student interaction are used by student modeling techniques, which provide estimates of student's knowledge state. This estimate

Faculty of Informatics, Masaryk University Brno; pelanek@fi.muni.cz, Phone +420-549 49 6991, Fax +420-549 491 820

is the basis for personalization algorithms. Research in student modeling focuses mainly on the intricacies of modeling temporal dynamics of learning and complex relations among knowledge components (Desmarais and Baker, 2012; Pelánek, 2017). In most cases, student modeling techniques consider only binary information about the correctness of answers. However, richer information about student behavior can be easily collected. Previous research considered such sources of richer information and their usage, e.g., a partial credit based on students' use of hints (Wang and Heffernan, 2013; Van Inwegen et al., 2015), high speed high stakes rule based on response times (Klinkenberg et al., 2011), or classification of incorrect answers based on the frequency of errors (Pelánek, 2018). However, such approaches are not utilized in the current mainstream student modeling approaches (Pelánek, 2017). Another closely related use of student interaction data is domain modeling, e.g., estimating difficulty of items (Klinkenberg et al., 2011), measuring the similarity of items (Pelánek, 2019), or creating and refining the mapping of items to knowledge components (Barnes, 2005; Desmarais et al., 2014). These techniques are also currently based primarily on the correctness data and could be improved by using richer information about student interactions.

Moreover, summarising students' performance in more detail than just the simple binary correctness is useful not just for modeling purposes but also for providing feedback to students and teachers. Providing students with feedback on their performance after each item supports their motivation and helps them to build metacognitive skills (Bull and Kay, 2007). Providing teachers with aggregated feedback on the performance of their students gives them information about what the whole class needs, as well as what the individual students need (Molenaar and Knoop-van Campen, 2017).

Using only correctness of answers is one extreme. The other extreme is to try to utilize all available information in an optimal way for a particular application; such approach has been used by Baker et al. (2012) to model student affective states and by Wang et al. (2017) to model knowledge states in programming. Utilizing detailed information about students' interaction with an item is potentially powerful, but significantly complicates the practical development of adaptive learning systems. Many different types of items are used in adaptive learning systems, and each of them produces specific interaction data. The approach based on the use of specific data requires for each application complex research into suitable modeling of students' performance.

We propose to use a compromise: answer classification, which can be seen as an interface between observations about student performance and various applications of these data in learning systems. Fig. 1 illustrates the main idea of this approach. The concept of interface is used widely in computer science. Its main advantage is that it facilitates the design and implementation of systems by decoupling the processing of raw data and algorithms that use the data. This is useful both from the development perspective (e.g., ease of development, maintainability) and research perspective (e.g., reproduction studies, evaluation of generalizability of results).

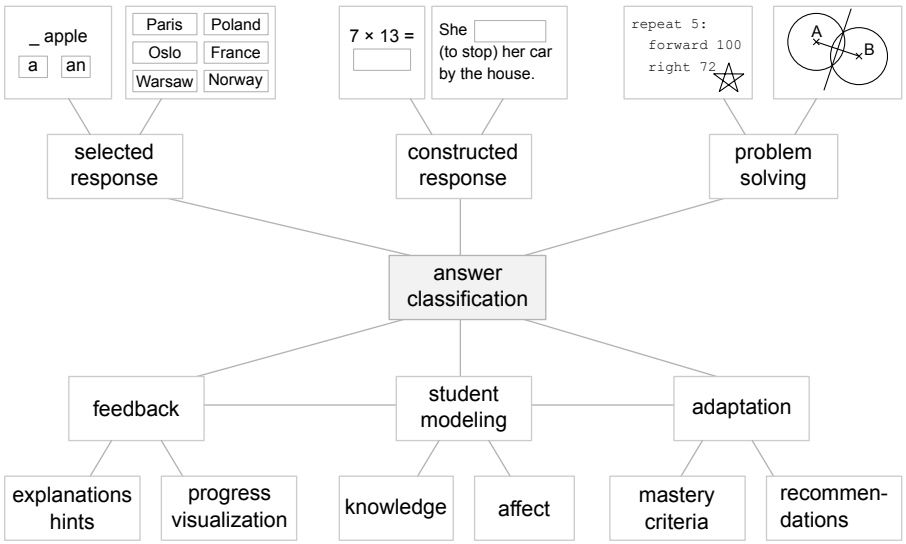

Fig. 1 Answer classification as an interface between data from exercises and algorithms determining system behavior. Only selected, illustrative examples of exercises and algorithms are shown.

The general idea illustrated in Fig. 1 can be realized using a wide variety of answer classification approaches. As a specific proposal, we put forward a classification of answers into six categories: three categories of correct answers (excellent, normal, weak) and three categories for incorrect answers (near miss, normal, non-serious attempt). We argue that this should be enough to capture the main differences in students' performance. The classification is widely applicable—it can be used for very different types of exercises and types of applications (e.g., modeling student knowledge or affect, providing feedback, mastery criteria). We discuss specific methods for classifying answers for different types of exercises, ranging from simple multiple-choice questions to interactive problem solving exercises. We support our proposal by analysis of extensive data from real adaptive learning systems.

2 Related Work

We provide an overview of research utilizing additional data beyond the correctness of answers. We focus on data that can be easily collected and that are relevant in many different learning systems, specifically response times, values of incorrect answers, and hints usage. Many additional sources of data have been used in previous work, particularly for modeling student affect (Woolf et al., 2009). Examples are data collected using eye movement tracking, facial expression camera, pressure sensitive chair, skin sensors, or mouse movements. However, these data are not easily collected and often do not scale. Such data are currently used mostly for lab studies where the goal is to develop sensor-

free detectors that would be based on easily available data (Paquette et al., 2014).

A common source of data about student performance that is simple to collect in a computerized learning system is response time. The use of response times has been explored for a long time in the context of adaptive testing (Van Der Linden, 2009). In adaptive learning, they have been used for various purposes: student modeling (Klinkenberg et al., 2011), engagement tracking (Beck, 2005), scheduling the learning of declarative knowledge (Mettler et al., 2011), analysis of slip and guess behavior (Baker et al., 2008), modeling off-task behavior (Baker, 2007), and distinguishing good and wrong use of bottom-up hints (Shih et al., 2011). The signal present in response times is, however, noisy and depends on a particular setting. In problem solving activities, response time can be the primary measure of performance (Pelánek and Jarušek, 2015). In other cases, the predictive power of response times seems to be limited (Pelánek, 2018; Papoušek et al., 2015).

Another potentially valuable source of data about student state is a specific value of an incorrect answer. Incorrect answers are known to have a highly skewed distribution with a few common values (Pelánek and Řihák, 2016; Stephens-Martinez et al., 2017). Such common incorrect answers have been used to improve student knowledge modeling (Nam et al., 2017; Řihák and Pelánek, 2016) or affect detection (Wang et al., 2015). Analysis of incorrect answers has also been used for labeling of errors (McTavish and Larusson, 2014; Straatemeier, 2014), clustering learners (Merceron and Yacef, 2005), and propagating feedback (Piech et al., 2015b). The analysis of incorrect answers for constructed response items is also related to techniques for automatic short answer grading (Burrows et al., 2015).

Learning systems often offer students hints. In such cases, data about hint usage provide another potentially valuable source of information about students. Hint usage data have been used for modeling help utility (Beck et al., 2008; Inventado et al., 2018) and for students' control and hint seeking behaviors (Goldin et al., 2013; Aleven and Koedinger, 2000; Aleven et al., 2003). For student modeling, hints have been used to determine a 'partial credit' as an extension of the basic binary evaluation of answers (Wang and Heffernan, 2013; Van Inwegen et al., 2015; Ostrow et al., 2015). Partial credit models (also called polytomous models) are used in the item response theory, which is applied mainly in the context of adaptive testing, e.g., a partial credit extension of the basic Rasch model (Masters, 1982). Such models can be useful, for example, in the case of a multiple choice question with several distractors, each with a different plausibility (Drasgow et al., 1995; Gierl et al., 2017).

The above-discussed approaches mostly focus on some feature of student performance and provide a tailored model for that feature. Although many approaches use machine learning techniques to learn specific values of model parameters, the basic structure of a model is mostly determined by researchers. An alternative approach is to use deep neural networks, which aim to learn a student model from the raw data about student performance. This approach has been proposed as 'deep knowledge tracing' (Piech et al., 2015a). Initially,

the approach used only data about the correctness of answers, but it has been quickly extended to various other types of data (Zhang et al., 2017; Wang et al., 2017). The advantage of this approach is that it eliminates the need to hand-craft specific student models. It has, however, several disadvantages. The obtained models are not easily interpretable, and interpretability may be important in many educational applications, e.g., for open learner modeling or for ensuring predictable behavior of a learning system. The deep learning methods also need large data to be useful; at the current stage of research, it is not even clear how large the data need to be. Also, the prediction accuracy of this approach is not clear since some of the claims of the improvement were shown to be due to methodological issues in evaluation (Pelánek, 2018).

To summarise, there exists extensive research on the use of a specific aspect of student performance data for specific purposes, which clearly illustrates the utility of the data beyond correctness. However, the systematic use of such data either in research or practice is lacking. Some authors have proposed the combined use of several data sources, a specific example is Arroyo et al. (2014), who use data about response times and hint usage in a practically deployed instructional strategy. Such use of performance data, however, is not simple and thus not very widespread. Our aim is to make the use of such data easier by providing an "interface" between the raw data and different applications.

3 Answer Classification: Possibilities, Proposal, Uses

In this section, we discuss general approaches to processing student performance data, outline our proposal for answer classification, and discuss its applications. In the next section, we discuss specific methods for answer classification and support them by data analysis.

3.1 Overview of Possible Approaches

On a general level, we deal with the following data processing pipeline: the learning system collects data about student interaction with an item; the data are processed, and the processed data are used in applications (e.g., student modeling, domain modeling, feedback). Fig. 1 provides an illustration of this pipeline.

One extreme approach to this pipeline is to avoid any processing and use the *raw data* directly for specific applications. This approach has the advantage that we avoid the loss of any potentially useful information. However, this approach makes the development of learning systems very difficult. Each type of exercise produces different raw data. Any technique (e.g., student modeling, mastery criteria) developed for such data would be specific for a particular setting and not reusable.

Another extreme approach is to consider only the *correctness of an answer*, i.e., to ignore any other interaction data. This is the currently dominant

approach, e.g., most commonly used student modeling techniques use only correctness data (Pelánek, 2017). This approach facilitates the development of reusable techniques since any specifics of a particular exercise are abstracted away. However, it also leads to a significant loss of potentially valuable information.

Between these two extremes, there is a spectrum of other approaches. Instead of just the binary evaluation, we can use a *continuous performance score*, e.g., value in the interval [0, 1]. Such a measure can be extended to be *multidimensional* to better capture different aspects of student performance. It is possible to use two-dimensional interpretable representation (e.g., "quality of final result" and "speed"), or multidimensional embedding that is automatically computed by machine learning techniques (and thus not directly interpretable) as done, for example, by Piech et al. (2015b).

Another approach is to use *discrete classification*, i.e., to classify student interaction data into one of several classes (e.g., "great performance", "correct, but slow", "incorrect, but reasonable", "fast guessing") and then use these classes for specific applications. With this approach, we lose some nuances of the data, but with a well-designed classification, we may be able to lose only a little information and get significant simplification.

3.2 Proposed Classification Approach

We argue that a suitable approach for improving the current state-of-the-art in adaptive learning systems is to utilize the classification approach. The advantage of this approach is the simplicity of its use in applications as it is only a slight extension of the basic correctness. At the same time, with a suitable choice of classes, we can achieve a limited loss of information and good interpretability of data processing, particularly compared to embeddings computed by neural networks.

Before proceeding to a specific proposal for answer classification, we explicitly clarify the setting. An answer classification is a function that takes as an input a log data on a student's interaction with a particular item and outputs a classification. An answer classification can be seen as a model of a student's performance on an item. Note that there is a crucial difference with respect to "student modeling" as commonly used (e.g., in models like Bayesian knowledge tracing or Additive factors model)—student modeling techniques use data on a sequence of items and take into account temporal dynamics (learning) along the sequence, whereas in answer classification we consider student performance on a single item.

For the progress of the adaptive learning field (both the development and research), it would be advantageous to have answer classification as universal as possible since this would enable easy reuse of techniques like student modeling or mastery criteria. There is, of course, a trade-off: a more general classification leads to a higher loss of information and worse interpretability.

Table 1 Proposed classification: answer classes and their potential interpretations.

	Class	Comments, interpretation
$\mathbf{C}^+$	correct, excellent	very smooth performance; a student knows the topic very well; cheating, lucky guess
$\mathbf{C}^0$	correct, normal	expected performance; an adequate item for the student
$\mathbf{C}^-$	correct, weak	a struggling student; an item was probably hard for a student
$\mathbf{I}^+$	incorrect, near miss	a reasonable attempt; a student seriously tried to answer the question, but made some partial mistake
$\mathbf{I}^0$	incorrect, normal	a student tried to solve an item, but was not successfull; the attempt gives evidence of knowledge gaps
$\mathbf{I}^-$	incorrect, non-serious	student did not seriously try to answer; disengaged behaviour; "just looking"; a misclick; gaming the system; pure guessing

Our aim in this work is to find a suitable compromise: a classification that provides reasonable specificity for a wide range of student interaction data.

To this end, we propose a classification outlined in Table 1. The classification distinguishes three classes of correct answers and three classes of incorrect answers. The choice of the number of classes is based on the analysis of data from a wide range of exercises; this analysis is discussed in the next section. Particularly for incorrect answers, the three proposed classes are naturally supported by the data. More classes could be beneficial in some use cases, but they bring additional costs (e.g., parameters that need to be estimated) and in general, we do not have strong support for them.

3.3 Contextual Data

In this work, we focus on the classification of a single answer. Taking into account contextual data (data about preceding answers) can potentially improve the classification, particularly by clarifying the cases which currently have multiple interpretations, e.g., $\mathbf{I}^-$ could mean both "disengaged behavior" and "misclick". These cases are almost impossible to distinguish by considering data about a single answer. However, when we take into account the context, their distinction may be quite clear.

The contextual data can be definitely useful. The question is, how to use them. One approach would be to consider them as an input to the answer classification. Another approach, which we prefer, is to perform the classification without contextual data, and then potentially in a second step use the contextual data (already classified) to perform an appropriate refinement or clarification. This is an advantageous decomposition: the use of contextual data is to a large degree independent of specifics of a particular type of exercise, and thus it can be done in an exercise independent way on the level of classes.

Moreover, the use of contextual data leads to more fine-grained classes and these are hard to specify in a universal way. An appropriate choice of more fine-grained classes depends on a particular application (e.g., cognitive modeling, affective modeling, feedback to students).

For illustration, we just mention examples of several rules illustrating how the use of contextual data can lead to a more detailed classification of answers:

- Answer $\mathbf{I}^-$ in the middle of a sequence of $\mathbf{C}^+$ and $\mathbf{C}^0$ answers: probably a misclick, i.e., an unintended submit of an incomplete answer.
- A sequence of mostly $\mathbf{I}^-$ with few $\mathbf{C}^+$: probably a guessing behavior.
- Answer $\mathbf{I}^+$ in a sequence of $\mathbf{C}^0$ and $\mathbf{C}^-$ answers: probably a productive struggle (encouraging feedback may be appropriate).
- Answer $\mathbf{I}^+$ in a sequence of $\mathbf{I}^-$: the student is probably just guessing and one of the guesses was slightly more lucky than others (encouraging feedback is not appropriate).
- A sequence of mostly $\mathbf{I}^0$, $\mathbf{I}^+$: "wheel spinning" (Beck and Gong, 2013)—the student is struggling, but probably unproductively and would benefit from the practice of prerequisites.

3.4 Applications of the Classification

The aim of the classification is to serve as an interface between raw data observation and different applications of data (as depicted in Fig. 1). In the rest of the work, we focus on the computation of answer classes. Here, we briefly outline their applications.

3.4.1 Student Modeling

Student modeling techniques use data on student interactions to provide an estimate of a student's state, which is subsequently used to guide the adaptive behaviors of a learning system. Student modeling can estimate different types of states: cognitive, meta-cognitive, affective.

The most commonly used type of student modeling is for the estimation of cognitive state, i.e., modeling students' knowledge of the domain. Most currently used models consider only binary information about the correctness of answers. These models can be extended to use a more detailed classification. We discuss the two most commonly used families of student modeling techniques (Pelánek, 2017).

For Bayesian knowledge tracing and other Hidden Markov models, the extension means that we need to have "emission probabilities" for the six classes instead of just the two classes that are used in the basic version of the model. For the Bayesian knowledge tracing model, it may be particularly useful to distinguish the "slip parameter" (probability of making a mistake when the student is in the "known state") for the classes $\mathbf{I}^+$, $\mathbf{I}^0$, $\mathbf{I}^-$.

Logistic models are based on a mapping of a continuous skill to the probability of correct answer, which is modeled by a logistic function. We can extend

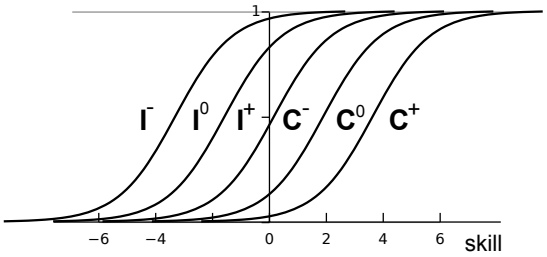

Fig. 2 Simple extension of a logistic student model with answer classification.

the model to consider several logistic functions, each of them specifying the probability of observing an answer of a specific type or better, see Fig. 2.

The classification can also be useful for modeling an affective state of students. For example, the $\mathbf{I}^-$ answers (particularly in a long uninterrupted sequence) are often evidence of an affective state than of a knowledge state.

3.4.2 Mastery Criteria

One common personalization approach in adaptive learning systems is mastery learning: students solve a sequence of items until they sufficiently master the practiced topic. To realize this approach, we need a mastery criterion—a procedure for deciding when to stop the practice. Previous research (Pelánek and Řihák, 2018) shows that for the purpose of determining mastery, the choice of input data is more important than the use of student modeling techniques. Even simple approaches like exponential moving average are sufficient when provided with suitable input data.

The proposed answer classification facilitates the application of mastery criteria in learning systems. The classification provides a sufficient summary of observed data and enables us to easily scale the implementation since we can use the same mastery criteria approach and formulas for many different types of exercises and topics.

3.4.3 Recommendations

Another personalization approach is recommending specific items, topics, or problems sets. This is often done based on the skill estimates provided by student modeling. In some cases, however, it may be useful to utilize answer classification for this purpose directly.

Consider an interactive problem solving exercise (e.g., a programming or multi-step mathematics problem) and suppose that a student started solving the problem but did not finish it. When the student returns to the system the next day, should the system recommend this unfinished problem or rather another one? In the case of the $\mathbf{I}^+$ answer, it is probable that the student tried hard to solve the problem and was not able to do so. Therefore, rather than recommending the same problem again, it is better to recommend another

one (preferably simpler). In the case of the $\mathbf{I}^-$ answer, the student did not seriously try to solve the problem (e.g., just glanced at the problem at the end of a session). Therefore, it is meaningful to recommend the problem for a repeated attempt.

3.4.4 Feedback to Students, Teachers, and Parents

Answer classification is also useful for providing feedback to students, teachers, and parents. Feedback is a crucial element in learning systems, and richer feedback has higher potential (Hattie and Gan, 2011). As a simple example, consider the case of incorrect answers. Acknowledging that the answer was $\mathbf{I}^+$ rather than $\mathbf{I}^0$ may have a positive psychological impact on students.

For teachers and parents, it is useful to have information about students progress. This information can be provided by the use of open learner models (Bull and Kay, 2007). These models provide a concise summary of student performance in the form of student skill estimates. However, for non-experts, it may be hard to understand precisely what is the meaning of these estimates and how are they computed. Presenting just a simple visualization of the answer classification data may be more understandable since the classes are interpretable and easy to understand.

3.4.5 Feedback to Developers and Content Authors

Answer classification can also be useful for providing feedback to developers and authors of the content. Development of practically used systems is often in line with what Baker (2016) calls "stupid tutoring systems, intelligent humans" and Aleven et al. (2016) call "adaptive design", i.e., a designer adapts a relatively simple design of the learning system based on a potentially nontrivial offline analysis of data.

Classification of answers is useful for many aspects of such offline educational data analysis. As a specific example, consider the estimation of item difficulty. For these estimates, it may be useful to filter out $\mathbf{I}^-$ answers, since these often correspond to users who are "just looking around" and do not reflect the real difficulty of items. Such answers can be distributed highly non-uniformly among items, and thus bias difficulty estimates for some items.

Another useful analysis is the ratio of $\mathbf{I}^+$ answers. A high ratio of $\mathbf{I}^+$ answers for some item is often an indication of a problematic item. It may be a poorly formulated item, e.g., an item permitting multiple valid responses which the author of the item did not consider, or an item assigned to a wrong knowledge component. Even if the item is valid, it may require specific attention concerning the preparation of explanations or hints.

4 Classification of Answers for Specific Types of Exercises

So far, we have provided a general discussion of the classification. Now we discuss specific types of exercises. We analyze a wide range of data to see whether

they provide natural support for the classification. Based on the analysis, we outline specific criteria for the classification of answers. We divide the discussion into three basic types of exercises that are commonly used in learning systems:

- *Selected response.* Students answer by selecting an answer from a provided choice.
- *Constructed response.* Students construct answers, typically by writing a number or a short text.
- *Interactive problem solving.* Students solve a problem in an interactive manner; the solution consists of a sequence of steps.

We focus on the basic analysis of individual aspects of student performance. The classification should support a variety of use cases, so it is important to consider and balance requirements imposed by many of them, and not to train an answer classifier optimized for a single usage, such as predictive power of a specific student model. Using the answer classification as feedback to students requires that not only the classes themselves but also the criteria for the classification are understandable. As a consequence, our aim is to define classes by simple criteria, rather than using machine learning techniques with many features.

4.1 Data and Methods

For the analysis, we use data from RoboMission (`robomise.cz`), an adaptive system for learning introductory programming, and from the Umíme learning system (`umimeto.org`, the adaptive practice of mathematics, English, Czech, and other educational domains for Czech students). The Umíme system contains many different types of exercises and offers the practice of fine-grained knowledge component. The adaptivity of the system consists of mastery learning (Pelánek and Řihák, 2018) and personalized recommendations. Within a specific exercise and knowledge component, specific items are presented to students in random order. The RoboMission system uses block-based programming in a microworld; the system uses adaptive recommendations of items to solve (Effenberger and Pelánek, 2018).

In order to make our analysis independent of the specific usage, we focus on the primary analysis of observed data. Specifically, we analyze relations between observed data for a single attempt (e.g., response time and the correctness of answers), and on the relation of the current performance to the future performance (e.g., the correctness or response time of the next attempt in a sequence).

For the analysis, we used normalized response times because untransformed response times are highly skewed and dependent on the specific presentation of a particular item (e.g., the length of a text). As a normalization, we transform raw response times into the "response time percentile", i.e., the percentage of other users that have a faster response time on the item.

4.2 Selected Response

In a selected response exercise, students select their response (answer) from a provided list of choices. A typical example is a multiple choice question, which uses just a few choices. This is one of the most widely used types of exercises for both assessment and learning. There are, however, other variants of selected response exercises, which provide students with a broader set of choices, e.g., categorization exercise, where the goal is to classify words into categories, pairing of matching words or expressions, or filling interpunction into a sentence.

4.2.1 Answer Properties

In the basic multiple choice questions where the answer is just a single choice, the answer cannot be analyzed in much detail. For incorrect answers, we may gain some information by taking into account a chosen distractor (Gierl et al., 2017), but for well-written questions this provides rather limited information. After an incorrect answer, a learning system may give students feedback that the answer was incorrect and provide them another opportunity to answer (Butler et al., 2007). In such a case, we may use the number of attempts necessary for finding the correct answer for the classification.

For more complex selected response exercises, an answer consists of several choices, e.g., tagging all nouns within a sentence. In these cases it may be useful to differentiate incorrect answers with respect to the number of mistakes within the item (e.g., the number of words tagged incorrectly within a sentence). A natural candidate for the $\mathbf{I}^+$ answer category (near miss) is the case of a single mistake. Analysis of our data suggests that this simple classification is surprisingly useful: it provides a balanced division of incorrect answers and it has good predictive ability.

Fig. 3 provides specific results for two exercises: a tagging exercise, where the goal is to tag words in a sentence (e.g., part of speech) or letters in a word, and a categorization exercise, where the goal is to sort words into correct categories (e.g., types of pronouns). The figure shows the predictive ability of different answers for next problem correctness—the results for "one mistake" class are robustly between the "correct" and "other incorrect" classes. In all these cases "one mistake" comprises between 40% and 70% of all incorrect answers, i.e., in all cases, it provides a reasonably balanced division of incorrect answers.

4.2.2 Response Times

To explore the usefulness of response times, we analyze its relation to correctness of answers. Fig. 4 shows results for the simplest multiple choice exercises with a choice from 2 options. The top row shows the relation to the current answer for the selection of several knowledge components. The bottom row

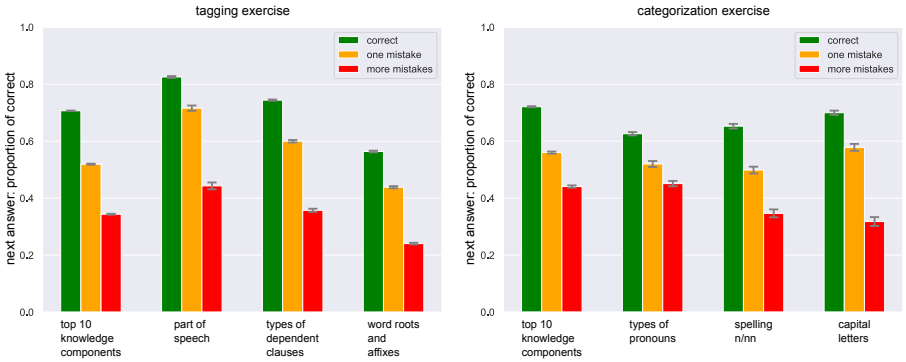

Fig. 3 Analysis of predictive power of answers (depending on the number of mistakes) for selected response exercises. Error bars show 95% confidence intervals.

shows the proportion of correct answers in the next step, divided according to the correctness of the current answer.

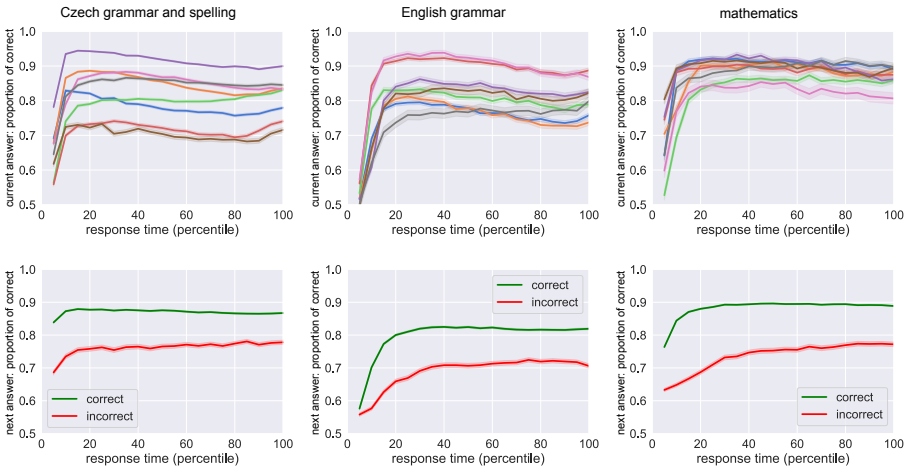

Fig. 4 Analysis of the relation between the normalized response time and the correctness of answers for multiple choice questions with one distractor (i.e., 50% guessing chance). Top row: the relation to the current answer for several knowledge components. Bottom row: the relation to the next answer separated according to the correctness of the current answer.

The basic results show are quite consistent across a variety of domains and knowledge components; similar results were also reported in previous work (Beck, 2005). The information present in response times is noisy and quite limited, with the exception of (very) fast responses. Under a specific threshold, we see a decreased proportion of correct answers for both the current and next answers: this probably corresponds to guessing behavior. Over this threshold, the curves are quite flat with only a minor trend. If the student answers correctly, then with higher response times, the proportion of corrent

Table 2 Proposed classification of answers for selected response exercises.

$\mathbf{C}^+$	very low response time (under 5th percentile)
$\mathbf{C}^0$	default correct
$\mathbf{C}^-$	use of hints (if present)
$\mathbf{I}^+$	one mistake (in case of multiple selections), second attempt correct
$\mathbf{I}^0$	default incorrect
$\mathbf{I}^-$	many mistakes, or very low response time (under 5th percentile)

answers in the next step very slightly decreases. On the other hand, if the student answers incorrectly, then with higher response times, the proportion of correct answers in the next step slightly increases. These trends are, however, small and probably of limited practical use.

The value of the threshold on very fast answers (guessing) is generally between the 5th and 10th percentile of response times. The exact value and the strength of the effect of the very fast answers depend on a specific exercise and knowledge components.

4.2.3 Proposed Classification for Selected Response Exercises

Based on the presented analysis and discussion, we propose a basic approach to the classification of answers for selected response exercises in Table 2.

For the class $\mathbf{C}^+$ we do not see any support for the classification with the interpretation "excellent". We see support for the classification based on fast response times with the interpretation "suspiciously good performance" (potentially obtained by guessing; in some cases, this may also be some form of cheating).

The class $\mathbf{C}^-$ can be naturally used in exercises with hints (a correct answer after taking a hint). Another intuitive candidate for this class is "high response time". However, we do not see any specific support for such classification in our data and we thus do not recommend the use of high response times for answer classification, unless supported by analysis for a particular application.

The class $\mathbf{I}^+$ can be naturally used for exercises with compound choice or repeated answers after mistakes. In this case, we see strong and robust support for a simple heuristic "exactly one mistake".

The class $\mathbf{I}^-$ can be classified by very low response times (pure guessing). A reasonable default for the choice of threshold is 5th percentile of response times. Another natural heuristic for this class is "many mistakes", but here the choice of the threshold depends on the type of exercise.

4.3 Constructed Response

With the constructed response format, students have to construct a response on their own, typically by writing. This leads to more opportunities for classification since we observe richer data. Specifically, there is now much wider

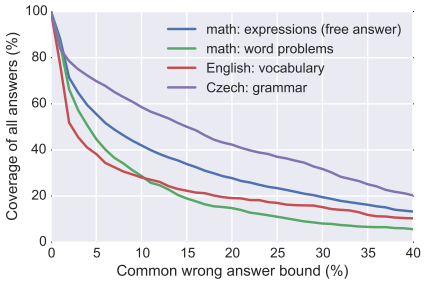

Fig. 5 Coverage of incorrect answers for different values of the bound that specifies which incorrect answers are considered "common".

freedom in "how to be incorrect". The interaction is also slower than for selected response and thus response times more varied.

4.3.1 Incorrect Answers

Previous research has repeatedly shown that the distribution of incorrect answers is highly skewed, i.e., for most items, few typical incorrect answers are covering most student mistakes (Pelánek and Řihák, 2016; Wang et al., 2015). Our data confirm this pattern. Typically the most common incorrect answer comprises 15–20% of all incorrect answers for a particular item. In some cases, the ratio can be even over 70% (examples from mathematics: an item $12-6+4$ and an answer 2, an item 4^2 and an answer 8). Using only the most common answer for classification could, however, be too limiting. On the other hand, a detailed analysis of many incorrect answers would lead to methods that are complex and hard to generalize. As a compromise, we explore a simple division of incorrect answers between "common" and "uncommon".

The simplest way to classify incorrect answers into common and uncommon is to consider as a common incorrect answer any answer which comprises at least $B\%$ of all incorrect answers. The question is how to choose the threshold B. To explore this question, we analyzed coverage of common incorrect answers for different settings of this bound. Fig. 5 shows the results for four types of exercises. For the analysis, we consider only items that have at least 50 incorrect answers. Based on this analysis, we suggest a bound 10% for the classification of an answer as a common incorrect answer. With this bound, common incorrect answers comprise between one third and one half of all answers.

Fig. 6 shows the analysis of the predictive power of answer types for several knowledge components in mathematics. We see that there is a consistent difference between common and uncommon incorrect answers—students who give a common incorrect answer are more likely to answer the next question correctly. Although the exact difference between the answer types depends on a particular knowledge component, the results are to a large degree consistent across a wide range of knowledge components. In most cases, the common

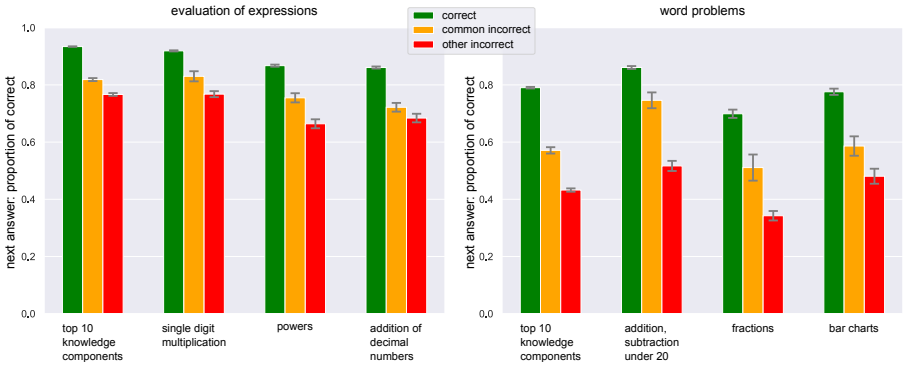

Fig. 6 Analysis of predictive power of answers (depending on the type of answer) for constructed response exercises. Error bars show 95% confidence intervals.

incorrect answers are close to the middle of the difference between correct answers and uncommon incorrect answers.

4.3.2 Response Times

With respect to response times, Fig. 7 presents analysis analogical to the analysis for selected response exercises (Fig. 4). The main difference is that in the case of constructed response items, it is very unlikely to obtain a correct answer by pure guessing. Consequently, the graphs for correct answers do not contain the slump for very fast responses. In fact, for correct answers the relation of response times to future performance is nearly linear with a very small negative slope, i.e., slower response means worse future performance. This effect is, however, minimal and probably not very useful for student modeling. For incorrect answers, the relation is again inverse, i.e., slower response times signals higher performance. For word problems, this effect is nontrivial. Nevertheless, the difference between different types of answers (common vs. uncommon) is more important.

We have also analyzed the relation between response time and future response time. In this case, the result shows positive, linear relation, i.e., normalized response times are relatively stable (slow students will remain slow).

4.3.3 Proposed Classification for Constructed Response Exercises

Based on the presented analysis, we propose a basic approach to the classification of answers for constructed response exercises in Table 3.

For correct answers, the analysis of our data does not show support for any simple general methods for distinguishing classes $\mathbf{C}^+$, $\mathbf{C}^0$, $\mathbf{C}^-$. The class $\mathbf{C}^-$ can be naturally used for exercises with hints and potentially can be specified using specific exercise data, particularly when high granularity data are available, e.g., when we collect data on the typing of individual letters, we may detect that a student is unsure since he wrote an incorrect answer

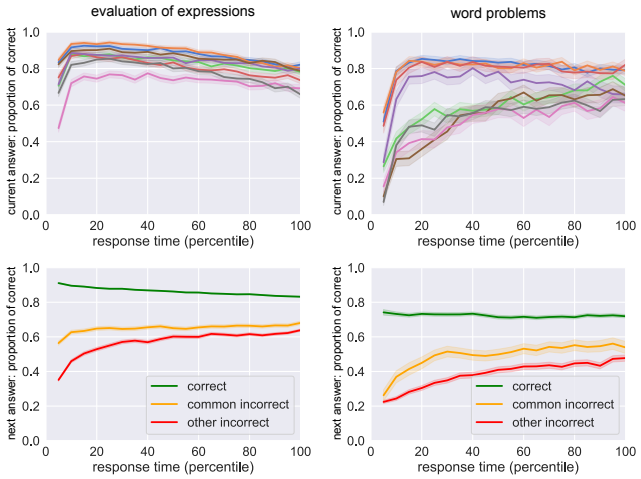

Fig. 7 Analysis of the relation between the normalized response time and the correctness of answers for constructed response exercises in mathematics. Top row: the relation to the current answer for several knowledge components. Bottom row: the relation to the next answer separated according to the correctness of the current answer.

Table 3 Proposed classification of answers for constructed response exercises.

$\mathbf{C}^+$	potentially low response time or application dependent data (fluency of answer)
$\mathbf{C}^0$	default correct
$\mathbf{C}^-$	use of hints, specific data
$\mathbf{I}^+$	common incorrect answer, small edit distance, second attempt correct
$\mathbf{I}^0$	default incorrect
$\mathbf{I}^-$	empty answer, wrong data type, very low response time

and then deleted it. In learning applications which aim at fluency, it may be sensible to distinguish $\mathbf{C}^+$ answers based on response times. However, our analysis does not support any simple choice of a threshold for classifying $\mathbf{C}^+$ answers.

For incorrect answers, the situation is different. For the class $\mathbf{I}^+$, we see strong support for the use of methods based on common incorrect answers. Such an approach is natural and widely applicable. Additional methods that are more exercise specific may also be useful, e.g., "small edit distance" (between the provided answer and the correct answer) or "second attempt correct".

For the class $\mathbf{I}^-$, we again see support for the criterion based on low response times, with the threshold between 5th and 10th percentile of response times for a particular item. In many cases, additional natural criteria can be used, particularly, "empty answer" and "wrong data type" (e.g., a student's answer is textual when a number was expected).

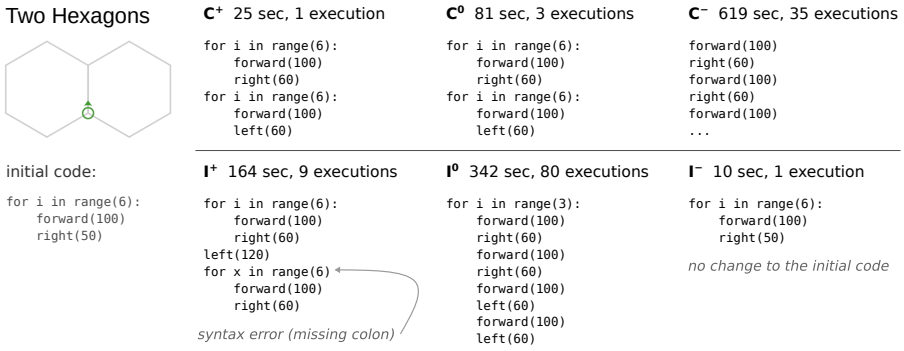

Fig. 8 Example of an item and answers in Turtle Python programming exercise. For each answer we provide summary statistics and the final submitted code.

4.4 Problem Solving

Problem solving encompasses a wide range of activities. We restrict our attention to well-structured problems for which automated support for students can be provided. Specifically, we consider interactive environments for answer construction with automated checking of correctness. Examples of such problem solving activities include logic puzzles, programming tasks, and geometry construction problems.

Interaction with the environment consists of a sequence of actions. For example, in programming exercises, the actions could be code edits, executions, and submits. To be consistent with the rest of the paper, we use the term *answer* to denote the whole time series of actions (not just a single action or just the final constructed product). Correspondingly, the term *correct answer* means that the student eventually solved the problem.

The structure of the answer is more complex than of the selected or constructed responses, and it varies wildly across problem solving exercises. We provide analysis and discussion specifically for introductory programming exercises. We analyzed data from four types of exercises: Python (programming with a textual output), Turtle Python (turtle graphics in Python), Turtle Blockly (turtle graphics in Blockly), and RoboMission (Effenberger and Pelánek, 2018). Fig. 8 provides an example of a Python Turtle item together with manually selected and classified examples of different answers.

4.4.1 Response Time and Performed Actions

To abstract from the details of a specific exercise, the time series of actions can be aggregated into a single number related to the performance, such as the total response time, or the number of performed actions. These two measures are nearly always available, so they are natural candidates for the classification.

The response time and the number of actions are related, but not perfectly. In four programming exercises we analyzed, the correlation was moderate and

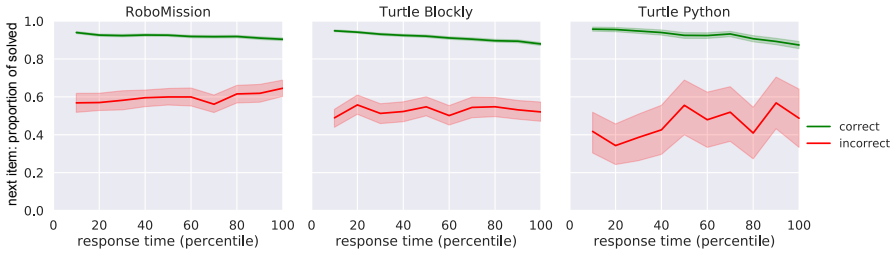

Fig. 9 The relationship between the response time and the proportion of successes for the next item in three programming exercises. The shaded areas show 95% confidence intervals.

dependant on the granularity of actions: the higher the granularity, the higher the correlation. In the exercises where either all code edits or all executions are logged, the correlation between time and actions was moderate for correct answers (Spearman's $r \geq 0.5$ for most items) and high for incorrect answers ($r \geq 0.7$ for most items). In the exercise where only submits are recorded, the correlation was low ($r \leq 0.4$ for all items).

Fig. 9 shows the relationship between the response time and the performance on the next item. For the correct answers, the higher response time is associated with a lower probability of solving the next item. This trend is consistent across exercises and individual problem sets. However, the effect size is rather small, especially in RoboMission, where the response time is used for a personalized recommendation, leading to a more difficult next item for students with a lower response time. Furthermore, the relationship is linear, without any natural thresholds for partitioning the answers into discrete classes. A similar relationship is between the number of performed actions and the probability of solving the next item.

Although an inspection of incorrect answers revealed some clearly non-serious answers with very low response time and just a few performed actions (as illustrated in Fig. 8), the overall relationship between the response time and the future performance is not robust (Fig. 9). This result suggests that the response time alone might not be enough to separate serious answers from non-serious, or that the number of non-serious answers is low for most of the items.

4.4.2 Interaction Network Path

The response time and the number of performed actions are not the only measures that can be derived from the series of the student's actions. The answer classification can utilize the whole *interaction network*, i.e., all actions of all students (Eagle et al., 2015). For example, the student's answer (a path in the interaction network) can be classified as either typical or atypical; a "divergent" incorrect path that is not similar to previously observed paths is probably an off-topic behavior (a non-serious answer).

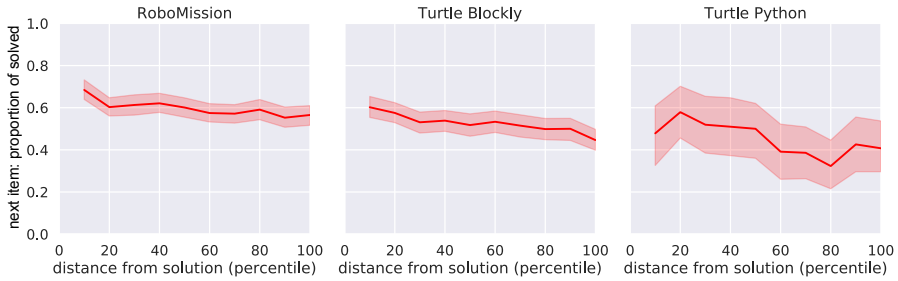

Fig. 10 The relationship between the minimum edit distance from a correct solution and the proportion of successes for the next item in three programming exercises (incorrect answers). The shaded areas show 95% confidence intervals.

On the other hand, if the student was close to a correct solution at one point (not necessarily as the last step), the answer could be considered as a "near miss" ($\mathbf{I}^+$). Fig. 10 shows the relationship between the minimum edit distance from any code constructed during the problem solving to a correct solution to the problem. The lower the edit distance, the higher the probability of solving the next item, which supports the hypothesis that students who were close to a correct solution have higher skills.

4.4.3 Quality of Solution

In addition to the qualities of the solving *process*, the answer classification can also consider the qualities of the final *product*. What are the important aspects of the product depends on the domain, e.g., in introductory programming, it can be functionality and style of the submitted program.

A natural measure of functionality of incorrect programs is the proportion of predefined test cases that have passed. This is, however, inapplicable in many introductory programming exercises, in which the students are asked to create a program without any input, e.g., to draw a specific shape in the turtle graphics. Instead, measures based on the edit distance from a correct solution (as in Section 4.4.2) might be used, which also have the advantage of not limiting themselves to syntactically correct programs.

For correct programs, the quality of the programming style can be evaluated. For example, an unapplied iteration or function abstraction indicate a gap in the student's knowledge. In introductory programming exercises, the information in source code is limited due to short programs and a limited number of available programming constructs. In this case, a sufficient heuristic for the style might be the length of the code, which is higher if the student's solution contains some repeated or redundant code.

We analyzed the power of the number of lines of the correct programs. The probability of solving the next item slightly decreases with the increasing length of the solution, but the overall trend is weak, especially in the easiest programming problems with extremely low variability of solutions. Nev-

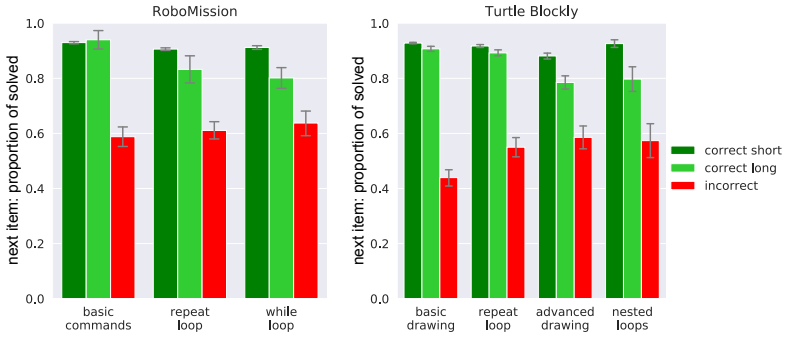

Fig. 11 Predictive power of the length of correct answers in programming exercises. The solutions are considered short if they have at most 20% more lines than the most common solution to the problem. Error bars show 95% confidence intervals.

ertheless, the relationship between the code length and future performance is stronger in more advanced problem sets, as shown in Fig. 11.

4.4.4 Proposed Classification for Problem Solving Exercises

Based on the analysis, we summarise our proposal to the classification of answers in problem solving exercises in Table 4. As we have only analyzed data from introductory programming exercises, it may be useful to perform a detailed analysis of the particular exercise considered.

As a general guideline, we propose to use the response time for the classification of correct answers. In problem solving, the response time is arguably an important aspect of the performance, it is readily available and it is the most universal. However, while the extreme answers are easy to classify (see Fig. 8), it is not clear how to set the thresholds. One approach, which we elaborate in Effenberger and Pelánek (2019), is to impose a constraint on the threshold variation between nearby problem sets, and iteratively improve the thresholds using a few manually labeled answers around the thresholds.

In some cases, there might be other natural candidates for the class $\mathbf{C}^-$ in addition to the response time. For example, if the exercise provides hints, or if the quality of the solution might be reasonably estimated using code length.

The class $\mathbf{I}^+$ can be used for incorrect answers that are close to a correct solution. In introductory programming exercises, the closeness can be measured by edit distance from a correct solution, and in more advanced programming exercises by the proportion of passed unit tests.

The class $\mathbf{I}^-$ might be detected by very low response time or very few performed actions, but our data do not provide support for any obvious choice of a threshold. Edit distance from a correct solution or a combination of multiple aspects might be useful.

Table 4 Proposed classification of answers for problem solving exercises.

$\mathbf{C}^+$	low response time, few unnecessary actions
$\mathbf{C}^0$	default correct
$\mathbf{C}^-$	high response time, many unnecessary actions, use of hints, long solution
$\mathbf{I}^+$	close to a solution (most tests passed, low edit distance from a solution)
$\mathbf{I}^0$	default incorrect
$\mathbf{I}^-$	very low response time, few actions, large distance from a solution

5 Discussion

We propose a classification framework that distinguishes six classes of answers in learning systems—three types of correct answers and three types of incorrect answers). We argue that this approach provides a suitable interface between raw performance data and diverse applications of such data (e.g., student modeling, recommendations, feedback). This proposed classification approach is not meant to be a universally optimal way to process answers; instead, it is a pragmatic compromise that should facilitate scalable development of adaptive learning systems. To conclude, we summarise our proposal, describe an example of an application, and outline directions that require further research.

5.1 Classification into Six Classes

The proposed approach uses six classes of answers. We have discussed specific classification criteria for commonly used types of exercises. The degree of support and usefulness of individual classes of answers depends on the type of exercise.

For exercises with simple interaction (i.e., selected and constructed response), differentiation of correct answers does not seem very natural and particularly useful, unless the exercise contains hints. It is possible to use response times for differentiating correct answers, but we do not see any strong support for this in data. Moreover, the response time has a strong implicit effect in the learning—students who answer quickly and correctly simply solve more items (within a given time) and thus achieve better performance. Using response time for the classification is therefore probably useful only if building fluency is an explicit goal of the exercise.

Classification of incorrect answers, on the other hand, can be often done in a natural way. Near miss answers (class $\mathbf{I}^+$) can be determined based on the structure of answer (compound answer with one partial mistake; edit distance to correct solution) or based on analysis of common incorrect answers. Note that when both of these approaches are applicable, they often strongly correlate—common incorrect answers are often those with exactly one mistake. For this classification approach, we have very robust support in our data analysis results. Non-serious answer (class $\mathbf{I}^-$) can be determined based on fast response times, with a suitable default threshold being the 5th percentile of response times.

For problem solving exercises, the full categorization is usable and more practically important. Consider a case of a simple programming exercise and two students who solved the exercise correctly: one after 20 seconds, another after 5 minutes. For each of them, a different recommendation of a follow-up activity is clearly warranted, i.e., an adaptation based only on the correctness of answer would be insufficient. The spectrum of performance is, however, in the case of problems solving rather continuous. It is not clear what aspects of performance and what thresholds to use for the classification. Problem solving exercises are also more varied compared to selected and constructed response exercises. The classification criteria are thus more dependent on the specific case. We have illustrated the analysis of data for the case of programming exercises; a similar approach can be used for other types of data.

5.2 Example of Application

To provide a specific illustration of the usage of the classification and of its advantages, we describe its application in the commercial learning system Umíme (`umimeto.org`). This system contains a wide variety of exercise types and focuses on the practice of individual knowledge components using principles of mastery learning. The original version of the used mastery criterion used binary information about answer correctness processed by an exponential moving average; see Pelánek and Řihák (2018) for details. One aspect of the used approach is that a small mistake just before reaching mastery leads to a significant drop in the progress bar—this behavior is quite frustrating for students.

We have extended the used mastery criterion to take into account the answer classification approach proposed in this paper. Specifically, the new version differentiates between different types of wrong answers, e.g., the progress towards mastery is penalized significantly more for "guess" than for "near miss". This differentiation leads to the introduction of new parameters into the mastery criterion (e.g., how large should be the difference between the penalization for "guess" and "near miss"?). The usage of the classification significantly simplified the design of the extended mastery criterion and setting of the parameters. It also leads to a modular and maintainable implementation.

5.3 Classification Usage

The proposed categories can be interpreted and used in different ways. In some applications (like feedback to students) it may be sufficient to consider only ordinal relations among classes. For many student modeling approaches, however, we need to assign classes some numerical values. Do we assign them values uniformly distributed between 0 and 1? The difference between $\mathbf{I}^+$ and $\mathbf{C}^-$ should be probably larger than the difference between $\mathbf{C}^-$ and $\mathbf{C}^0$, i.e., we may want to assign classes specific values between 0 and 1. Is there any reasonably universal assignment of such values, or do we need to set them specifically for each application? These questions need further research.

One of the main applications of performance data is the modeling of student knowledge. As we have argued, current student modeling techniques focus on the use of binary correctness data, but they can be quite directly extended to use a discrete answer classification. In this work, we evaluated the proposed classes using it as a trivial student model predicting next problem correctness. Incorporation of the classification into student modeling techniques can increase or decrease the usefulness of individual classes. These interactions also require more research.

5.4 Limitations and Extensions

One aspect of learning systems that we covered only briefly and that can significantly influence the classification is the presence and use of hints. The availability of hints should clearly be taken into account in the design and interpretation of a classification for a specific exercise. It is, however, difficult to discuss hints in more detail on a general level as hints differ widely in their content (from a weak hint that guides students to the right direction to a strong hint that describes all details of a solution) and their availability (e.g., on demand, automatic after an incorrect answer, or dynamically based on a student model).

In this work, we consider classification based on data about a single answer. Contextual data (data about a whole sequence of answers) can be useful for distinguishing random guessing from misclicks or fluent, correct answers from lucky guesses (as outlined in Section 3.3). Such use of contextual data can be done as a second step, using a sequence of answers classified into the proposed six classes. Such a two-step approach leads to some loss of information, but it is simpler and more easily generalizable. The use of contextual data for answer classification, however, needs more research.

Instead of classifying just single answers, it would be possible to classify student's performance over a whole sequence of answers, i.e., performance on a knowledge component. The basic approach to such classification is again binary—in this case into "mastered" and "not mastered". A more nuanced classification would be useful for sequencing and recommendation of knowledge components, specifically for spaced repetition of practice: if a student mastered a knowledge component with a struggle, it makes sense to propose an early repetition of practice; if a student mastered a knowledge component without any problems, repetition of practice is probably not necessary. Such classification could be done in a similar way to the classification presented in this work.

References

Aleven, V. and Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? In *Proceedings of Intelligent Tutoring Systems*, pages 292–303. Springer.

Aleven, V., McLaughlin, E. A., Glenn, R. A., and Koedinger, K. R. (2016). *Handbook of research on learning and instruction*, chapter Instruction based on adaptive learning technologies, pages 522–559. Routledge.

Aleven, V., Stahl, E., Schworm, S., Fischer, F., and Wallace, R. (2003). Help seeking and help design in interactive learning environments. *Review of educational research*, 73(3):277–320.

Arroyo, I., Woolf, B. P., Burelson, W., Muldner, K., Rai, D., and Tai, M. (2014). A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. *International Journal of Artificial Intelligence in Education*, 24(4):387–426.

Baker, R. S. (2007). Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1059–1068. ACM.

Baker, R. S. (2016). Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, 26(2):600–614.

Baker, R. S., Corbett, A. T., and Aleven, V. (2008). More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Proceedings of Intelligent Tutoring Systems*, pages 406–415. Springer.

Baker, R. S., Gowda, S. M., Wixon, M., Kalka, J., Wagner, A. Z., Salvi, A., Aleven, V., Kusbit, G. W., Ocumpaugh, J., and Rossi, L. (2012). Towards sensor-free affect detection in cognitive tutor algebra. In *Proceedings of Educational Data Mining*.

Barnes, T. (2005). The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, pages 1–8.

Beck, J. E. (2005). Engagement tracing: using response times to model student disengagement. In *Proceedings of Artificial intelligence in education*, volume 125.

Beck, J. E., Chang, K.-m., Mostow, J., and Corbett, A. (2008). Does help help? introducing the bayesian evaluation and assessment methodology. In *Proceedings of Intelligent Tutoring Systems*, pages 383–394. Springer.

Beck, J. E. and Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. In *Proceedings of Artificial Intelligence in Education*, pages 431–440. Springer.

Bull, S. and Kay, J. (2007). Student models that invite the learner in: The smili open learner modelling framework. *International Journal of Artificial Intelligence in Education*, 17(2):89–120.

Burrows, S., Gurevych, I., and Stein, B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25(1):60–117.

Butler, A. C., Karpicke, J. D., and Roediger III, H. L. (2007). The effect of type and timing of feedback on learning from multiple-choice tests. *Journal of Experimental Psychology: Applied*, 13(4):273.

Desmarais, M., Beheshti, B., and Xu, P. (2014). The refinement of a q-matrix: Assessing methods to validate tasks to skills mapping. In *Proceedings of*

Educational Data Mining.

Desmarais, M. C. and Baker, R. S. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38.

Drasgow, F., Levine, M. V., Tsien, S., Williams, B., and Mead, A. D. (1995). Fitting polytomous item response theory models to multiple-choice tests. *Applied Psychological Measurement*, 19(2):143–166.

Eagle, M., Hicks, D., Peddycord III, B., and Barnes, T. (2015). Exploring networks of problem-solving interactions. In *Proceedings of Learning Analytics And Knowledge*, pages 21–30. ACM.

Effenberger, T. and Pelánek, R. (2018). Towards making block-based programming activities adaptive. In *Proceedings of Learning at Scale*, page 13. ACM.

Effenberger, T. and Pelánek, R. (2019). Measuring students performance on programming tasks. In *Proceedings of Learning at Scale*. ACM.

Gierl, M. J., Bulut, O., Guo, Q., and Zhang, X. (2017). Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review. *Review of Educational Research*, 87(6):1082–1116.

Goldin, I., Koedinger, K., and Aleven, V. (2013). Hints: You can't have just one. In *Proceedings of Educational Data Mining*.

Hattie, J. and Gan, M. (2011). Instruction based on feedback. *Handbook of research on learning and instruction*, pages 249–271.

Inventado, P. S., Scupelli, P., Ostrow, K., Heffernan, N., Ocumpaugh, J., Almeda, V., and Slater, S. (2018). Contextual factors affecting hint utility. *International Journal of STEM Education*, 5(1):13.

Klinkenberg, S., Straatemeier, M., and Van der Maas, H. (2011). Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824.

Masters, G. N. (1982). A rasch model for partial credit scoring. *Psychometrika*, 47(2):149–174.

McTavish, T. S. and Larusson, J. A. (2014). Labeling mathematical errors to reveal cognitive states. In *Open Learning and Teaching in Educational Communities*, pages 446–451. Springer.

Merceron, A. and Yacef, K. (2005). Clustering students to help evaluate learning. In *Technology Enhanced Learning*, pages 31–42. Springer.

Mettler, E., Massey, C. M., and Kellman, P. J. (2011). Improving adaptive learning technology through the use of response times. In *Proceedings of Conference of the Cognitive Science Society*, pages 2532–2537.

Molenaar, I. and Knoop-van Campen, C. (2017). Teacher dashboards in practice: Usage and impact. In *Proceedings of European Conference on Technology Enhanced Learning*, pages 125–138. Springer.

Nam, S., Frishkoff, G. A., and Collins-Thompson, K. (2017). Predicting short- and long-term vocabulary learning via semantic features of partial word knowledge. In *Proceedings of Educational Data Mining*.

Ostrow, K., Donnelly, C., Adjei, S., and Heffernan, N. (2015). Improving student modeling through partial credit and problem difficulty. In *Proceedings of Learning at Scale*, pages 11–20. ACM.

Papoušek, J., Pelánek, R., Řihák, J., and Stanislav, V. (2015). An analysis of response times in adaptive practice of geography facts. In *Proceedings of Educational Data Mining*, pages 562–563.

Paquette, L., Baker, R. S., Sao Pedro, M. A., Gobert, J. D., Rossi, L., Nakama, A., and Kauffman-Rogoff, Z. (2014). Sensor-free affect detection for a simulation-based science inquiry learning environment. In *International Conference on Intelligent Tutoring Systems*, pages 1–10. Springer.

Pelánek, R. (2017). Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3):313–350.

Pelánek, R. (2018). The details matter: methodological nuances in the evaluation of student models. *User Modeling and User-Adapted Interaction*, 28:207–235.

Pelánek, R. (2018). Exploring the utility of response times and wrong answers for adaptive learning. In *Proceedings of Learning at Scale*. ACM.

Pelánek, R. (2019). Measuring similarity of educational items: An overview. *IEEE Transactions on Learning Technologies*.

Pelánek, R. and Jarušek, P. (2015). Student modeling based on problem solving times. *International Journal of Artificial Intelligence in Education*, 25(4):493–519.

Pelánek, R. and Řihák, J. (2018). Analysis and design of mastery learning criteria. *New Review of Hypermedia and Multimedia*, 24:133–159.

Pelánek, R. and Řihák, J. (2016). Properties and applications of wrong answers in online educational systems. In *Proceedings of Educational Data Mining*, pages 466–471.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., and Sohl-Dickstein, J. (2015a). Deep knowledge tracing. In *Proceedings of Advances in neural information processing systems*, pages 505–513.

Piech, C., Huang, J., Nguyen, A., Phulsuksombati, M., Sahami, M., and Guibas, L. (2015b). Learning program embeddings to propagate feedback on student code. In *Proceedings of International Conference on Machine Learning*, volume 37 of *ICML'15*, pages 1093–1102.

Shih, B., Koedinger, K. R., and Scheines, R. (2011). A response time model for bottom-out hints as worked examples. *Handbook of educational data mining*, pages 201–212.

Stephens-Martinez, K., Ju, A., Parashar, K., Ongowarsito, R., Jain, N., Venkat, S., and Fox, A. (2017). Taking advantage of scale by analyzing frequent constructed-response, code tracing wrong answers. In *Proceedings of International Computing Education Research*, pages 56–64. ACM.

Straatemeier, M. (2014). *Math Garden: A new educational and scientific instrument*. PhD thesis, Universiteit van Amsterdam, Faculty of Social and Behavioural Sciences.

Van Der Linden, W. (2009). Conceptual issues in response-time modeling. *Journal of Educational Measurement*, 46(3):247–272.

Van Inwegen, E. G., Adjei, S. A., Wang, Y., and Heffernan, N. T. (2015). Using partial credit and response history to model user knowledge. In *Proceedings of Educational Data Mining*.

Řihák, J. and Pelánek, R. (2016). Choosing a student model for a real world application. In *Proceedings of Building ITS Bridges Across Frontiers (ITS Workshop)*.

Wang, L., Sy, A., Liu, L., and Piech, C. (2017). Learning to represent student knowledge on programming exercises using deep learning. In *Proceedings of ducational Data Mining*, pages 324–329.

Wang, Y. and Heffernan, N. (2013). Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes. In *Proceedings of Artificial Intelligence in Education*, pages 181–188. Springer.

Wang, Y., Heffernan, N. T., and Heffernan, C. (2015). Towards better affect detectors: effect of missing skills, class features and common wrong answers. In *Proceedings of Learning Analytics And Knowledge*, pages 31–35. ACM.

Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, D., and Picard, R. (2009). Affect-aware tutors: recognising and responding to student affect. *International Journal of Learning Technology*, 4(3-4):129–164.

Zhang, L., Xiong, X., Zhao, S., Botelho, A., and Heffernan, N. T. (2017). Incorporating rich features into deep knowledge tracing. In *Proceedings of Learning at Scale*, pages 169–172. ACM.