JOURNAL OF LEARNING ANALYTICS

SOLAR SOCIETY for LEARNING ANALYTICS RESEARCH

Volume 8(2), 93–104. https://doi.org/10.18608/jla.2021.7345

Analyzing and Visualizing Learning Data: A System Designer's Perspective

Radek Pelánek[1]

Abstract

In this work, we consider learning analytics for primary and secondary schools from the perspective of the designer of a learning system. We provide an overview of practically useful analytics techniques with descriptions of their applications and specific illustrations. We highlight data biases and caveats that complicate the analysis and its interpretation. Although we intentionally focus on techniques for internal use by designers, many of these techniques may inspire the development of dashboards for teachers or students. We also identify the consequences and challenges for research.

Notes for Research

- We provide an overview of analytics and visualization techniques from the perspective of a system designer.
- Most of the presented techniques are also relevant for teachers and other stakeholders. Two interesting research questions are as follows: What is the best way to present this type of analysis to teachers? How can we make them actionable?
- We highlight biases and caveats in learning data. Understanding these issues is necessary in order to correctly interpret research results.

Keywords

Difficulty, similarity, response time, temporal patterns, biases, cheating

Submitted: 29/07/20 — **Accepted:** 25/05/21 — **Published:** 03/09/21

Corresponding author [1] *Email: pelanek@fi.muni.cz Address: Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic. ORCID ID: https://orcid.org/0000-0001-8877-4729*

1. Introduction

Learning analytics and educational data mining techniques are used in many different ways. One common application is feedback about the learning process: to teachers using teacher dashboards (Molenaar & Knoop-van Campen, 2017; Xhakaj, Aleven, & McLaren, 2017), to students using open student models (Bull & Kay, 2007), or to managers and policymakers using analytics to influence the educational system (Clow, 2012). Another use is for researchers who are designing and evaluating student models that guide the adaptive behaviour of learning systems (Pelánek, 2017).

In this work, we discuss learning analytics from the perspective of a designer of a learning system for primary and secondary schools. We use the term *designer* broadly, encompassing many people involved in the creation and maintenance of a learning environment. With respect to analytics, the designer's point of view is focused particularly on the iterative improvement of the system based on analyzing collected data. In previous work, this approach has been called "design-loop adaptation" (Aleven, McLaughlin, Glenn, & Koedinger, 2016) or "stupid tutoring system, intelligent humans" (R. S. Baker, 2016).

We consider the designer's perspective because it provides some unique points of view and it is also rarely discussed in published papers (as opposed to the other perspectives). Many of the techniques that we discuss are not unique to the designer's perspective, but this perspective can provide interesting inspiration, e.g., for the development of teacher and student dashboards. We also discuss some biases and caveats in data that are detected naturally from the designer's perspective but would be easy to miss from other perspectives (e.g., a researcher evaluating student models).

The paper is structured as follows. We introduce the specific learning system that is used for examples through the paper. The system is a typical case study of an adaptive practice system for primary and secondary schools. The core of the paper is an overview of analytics and visualization techniques; we discuss the purpose and variants of each technique and provide examples. We conclude with a discussion of biases and caveats that can influence the interpretation of the data analysis. This part is particularly important for researchers because ignoring caveats in datasets can lead to a misleading interpretation of the research results.

2. Context

In our discussion, we use the following terminology. We use *items* as a generic term for describing problems, questions, or exercises that students solve in a learning system. We assume that items are mapped into *knowledge components* (also called skills or concepts) (Koedinger, Corbett, & Perfetti, 2012; Pelánek, 2020a). We consider a learning system with content for primary and secondary schools and focus on simple items that mostly concern recognition and recall of factual knowledge or applying simple procedures. Although this may seem restrictive, such items are relevant to a large portion of primary and secondary education.

For the discussion, we use examples from the learning system *Umíme to*. The system is used by approximately 5% of Czech schools; its basic features are similar to systems available in English and other languages, e.g., Khan Academy (Ruipérez-Valiente, Muñoz-Merino, Leony, & Kloos, 2015), Math Garden (Klinkenberg, Straatemeier, & Van der Maas, 2011), ASSISTments (Heffernan & Heffernan, 2014), and IXL.

The system covers several learning domains, e.g., mathematics, language learning, and programming. It covers hundreds of knowledge components and includes tens of thousands of items. The system's functionality includes support for virtual classrooms (particularly for automatically evaluated homework), personalized recommendations, mastery criteria, and multiplayer games. The system can be used without a licence (with a limit on the number of answers per day) or with a school or family licence. It is used by approximately ten thousand students per day.

3. Analysis and Visualization Techniques

In this section, we discuss techniques for analyzing and visualizing learning data. We do not aim at completeness. We highlight particular techniques that we deem to be worthy of attention either because they are very useful for development or because they are underused and would benefit from more attention (from both practitioners and researchers). We provide specific examples for each technique and discuss its applications.

3.1 Item Difficulty and Discrimination

A key analysis for designers is the analysis of item difficulty. Also worthy of attention is the less studied but useful concept of item discrimination, i.e., how well an item discriminates between students who have different skills. These parameters have several applications in the development of a learning system:

- Improving the domain model (mapping of item and knowledge components): Typically, we want to have item sets with homogeneous difficulty (Pelánek, 2020a).
- Finding inappropriate items: Items that are too difficult, too easy, or non-discriminating are suspicious and should be checked.
- Feedback to content authors and impetus for new content development: Typical feedback based on the analysis is "This seems to be a difficult type of item. We need something similar but slightly easier."

Difficulty and discrimination are typically analyzed with respect to the correctness of answers. Difficulty can be most easily quantified using the success rate (the ratio of students who answered correctly). Discrimination can also be quantified using simple statistics, e.g., using the upper-lower discrimination index (Brennan, 1972). Consider a set of students who answered item i. We sort these students with respect to their overall performance on other items and take the upper third U and the lower third L. The discrimination index DI_i is the difference between the success rates on item i of students in sets U and L. The second simple method of discriminant calculation is the point biserial index PB_i, which is given as the Pearson correlation coefficient between the overall success rate of students (continuous variable) and their answer to item i (dichotomous variable).

A more systematic approach is to use item response theory models (De Ayala, 2013), which incorporate both difficulty and discrimination parameters and are able to take into account non-randomly missing data. However, basic item response theory models were developed for the purpose of testing and do not take learning into account. There are many student modelling techniques (Pelánek, 2017) that take learning into account, but under specific assumptions, and basic versions of these models do not have item difficulty parameters.

A natural question follows: For analysis, is it sufficient to use simple methods (like the success rate) to measure difficulty, or is it necessary to use more complex models? The answer to this question, of course, depends on specific data, on the learning

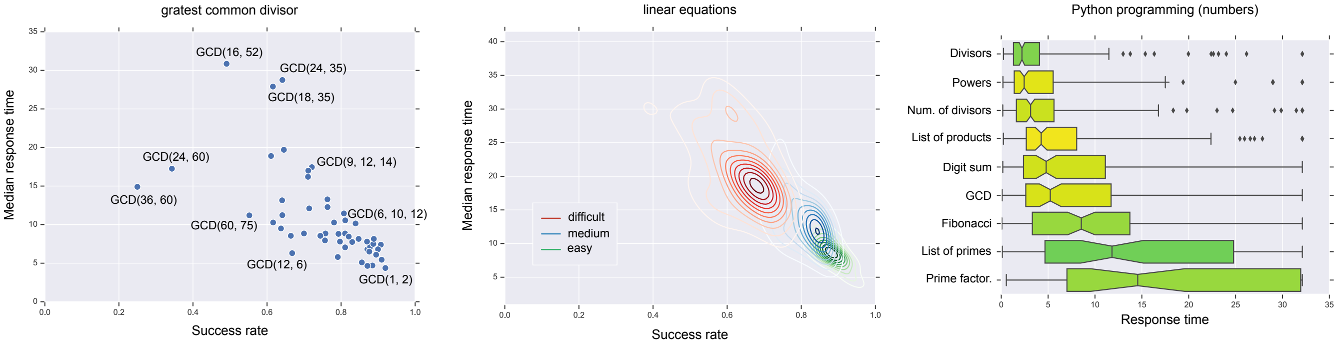

Figure 1. Three different visualizations of difficulty analyses based on the combination of the success rate and the response time.

system, and in particular on the presence of biases in the data (which we discuss in the next section). Our experience is that in many cases even simple methods provide sufficiently good estimates for the above-discussed applications; see Wauters, Desmet, and Van Den Noortgate (2012) for a more detailed discussion.

Pragmatically, rather than study complex models, it is often more useful to consider other aspects of performance than just the correctness of answers, in particular response times. Although response times are often correlated with correctness, the correlation is never perfect and response times often provide complementary and useful insight into item difficulty (see Van Der Linden (2009) for a conceptual discussion of response times in the context of testing). Figure 1 shows examples of three visualizations that combine correctness and response times, each with a different focus:

- The first graph is a basic scatter plot, which is useful in particular for noticing outlying items.
- The second graph is a kernel density estimate (KDE) plot, which has the same axes as the first graph but provides a more high level view. The depicted case shows a comparison of three sets of linear equation items. This visualization is particularly useful for checking whether items are appropriately distributed among item sets.
- The third graph displays not just the median response time, but the whole distribution of response times, as a boxplot. The success rate information is depicted as the colour of the boxplot. This visualization is particularly useful for sequencing items in problem-solving exercises.

These visualizations allow system designers to quickly notice problems with the content and possibilities for improvement; e.g., in the middle graph in Figure 1, "easy" and "medium" levels have non-trivial overlap, which can be improved by swapping items between these levels. We have successfully used these visualizations with a simple, yet efficient approach: creating an overview visualization, which contains a single type of visualization (e.g., the KDE plot) for all knowledge components in the system. Thanks to human sensitivity to patterns, a system designer can quickly identify problematic knowledge components by simply scanning such a visualization. When the number of knowledge components is too large for such a purely visual approach, it is possible to create heuristics that provide pre-filtering of suspicious content. One such approach that we have used is to compute an index expressing overlap of two difficulty levels and then sort the KDE plots based on this index.

3.2 Item Similarity

Learning systems contain a large number of items. In the case of our system, the number of items is on the order of tens of thousands. Managing such a large set of items is complex. A useful tool in this regard is the analysis of item similarity. Item similarity can be computed in two basic ways: based on the content of items (statement, solution, explanation) or on the performance of students (Pelánek, 2020b).

As a specific illustration, consider the following examples of simple similarity measures for two items i and j. A basic content-based method is to compute the normalized edit distance on problem statements s_i and s_j. A basic performance-based method is to compute the Pearson correlation coefficient of student answers on items i and j. This simple method can be used for any type of item and provides surprisingly useful results. Figure 2 shows 2D projections of items based on this similarity measure. Note that the content of items is used only for colouring the points; the projection itself is computed based only on the correctness of student answers. The first graph (one-step equations) shows that the similarity based on performance corresponds to operations used in equations. The other two graphs show examples from English grammar and provide useful insight. For indefinite articles, there is a clear group of outliers (words where the pronunciation is important for the choice of an article); this suggests that more examples and explanations of this type are needed in the system. For prepositions of time, we see that

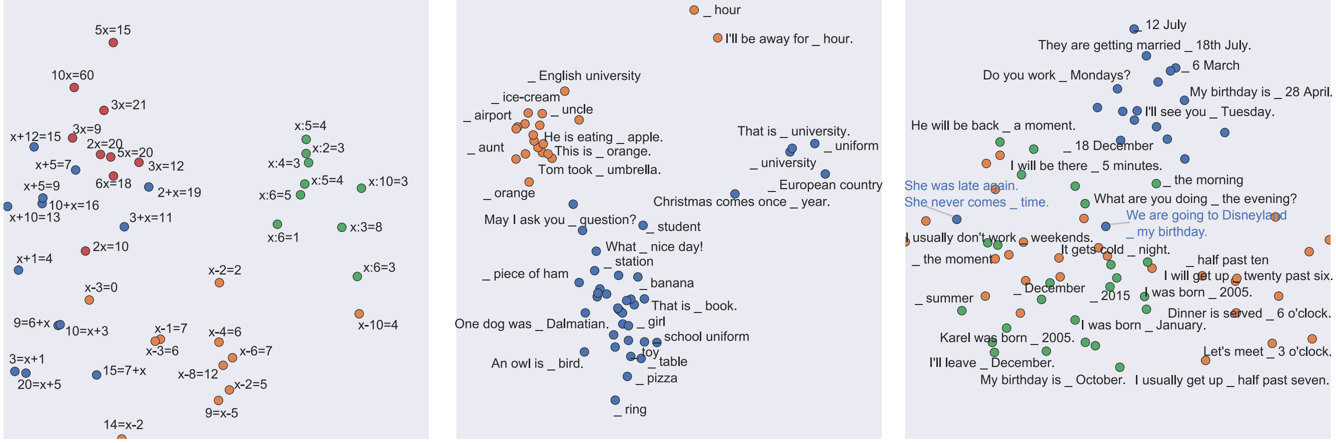

Figure 2. Projection of items based on similarity of student performance.

Table 1. Common wrong answers in English vocabulary practice

Type of Mistake	Expected Answer	Common Wrong Answer
Misspelling	friend	frend
	brother	brather
	brown	braun
Confusing semantically similar words	nephew	cousin
	wife	woman
	silver	grey
Alternative answer	fireman	firefighter
	trousers	pants
	grandfather	grandpa

items for the preposition "on" create a coherent cluster, with the exception of two items (highlighted in the figure). This again deserves the attention of content writers. In Figure 2, the basic PCA was used to project the similarities into the plane. By using a more advanced projection method (e.g., t-SNE), it is possible to obtain more detailed insight into item similarities.

This kind of visualization can serve many purposes: the identification of item clusters (which often suggests a division of items into separate knowledge components), the identification of outliers and duplicated (or too similar) items, or inspiration for "missing items." A specific, pragmatic application is the preparation of explanation and feedback messages. By using 1D projection based on similarity, we obtain a list of items, with similar items next to each other. Such a list significantly simplifies the preparation of explanations by content authors.

3.3 Common Wrong Answers

Wrong answers typically have a very skewed distribution; i.e., for most items, there are few common wrong answers that occur very often (Wang, Heffernan, & Heffernan, 2015; Pelánek & Řihák, 2016). Analysis of these common wrong answers is very valuable for system designers and has applications in several directions.

Answers that are extremely common (the same frequency as the correct answer or even higher) typically indicate some problem with an item—this can be a clear mistake in the item; an ambiguously formulated item that allows several interpretations; or a correctly formulated item used in the wrong context, which confuses students. Such cases typically require direct action (to delete, update, or move the item).

Other cases of common wrong answers may indicate student misconceptions during learning. The presence of common wrong answers may serve as a tool for the preparation of hints and explanations (Inventado et al., 2018). Common wrong answers can also be used to model student skills and mastery criteria (Pelánek & Řihák, 2018). Finally, common wrong answers can be used for automatic question generation—in selected response exercises (e.g., multiple-choice questions), we can create competitive distractors by using common wrong answers from constructed response exercises (Pelánek, 2020a).

Table 1 provides several examples of specific common wrong answers in English vocabulary learning. In this case, we can divide the mistakes into three main classes. Note that for each of these classes, it would be advantageous to provide students with a different kind of feedback from the system.

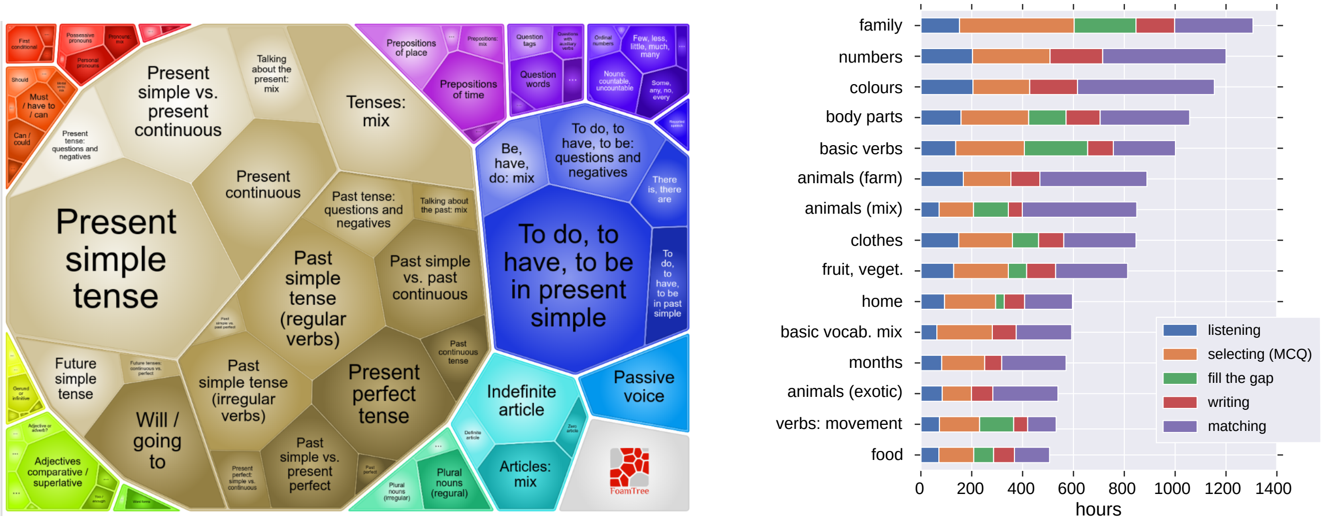

Figure 3. Visualization of the distribution of student activity. Left: Treemap visualization of the popularity of English grammar topics. Right: Vocabulary topics and distribution across exercise types.

3.4 Student Activity

Learning systems need to contain a wide range of topics and activities. This leads to the following questions: How is student activity distributed among the available content? Where do students spend their time?

The basic application of this type of analysis is to determine priorities for development. When a topic is popular, it is useful to extend the coverage or add explanations and hints. When students spend little time on a topic, it may indicate problems in the system (e.g., mistakes in the content or poor meta-data leading to incorrect navigation to the content). Figure 3 provides two examples of visualizations of student activity. The first graph shows a foam tree[1]—a specific variant of a treemap visualization. Treemaps provide a quick overview of overall student activity and are particularly useful for setting and checking priorities for the development of content. The second graph shows more specialized plot activity across different types of exercises for the practice of vocabulary. This graph not only shows which topics are most popular but also highlights that student activity differs quite widely for different topics. This observation leads to a reflection on possible reasons for these differences, which provides ideas for adding content or revising the user interface.

These types of activity analysis are not very specific to learning systems; similar analyses are relevant to any current web-based application. A more specific application of student activity data is the checking and revision of meta-data. An important part of the domain model is the mapping of knowledge components to grades, which is used, e.g., for navigation and personalized recommendation. This mapping should ideally be manually specified by an expert and based primarily on the standard curriculum in a given country. However, with hundreds of knowledge components, this mapping is expensive to perform and error-prone. Moreover, the reality of classrooms is not completely aligned with official curriculum recommendations. It is thus useful to perform automated checks—a simple yet useful analysis is to check the correspondence to popularity per grade (e.g., based on homework) and report the largest deviations from the manually specified grade recommendation. Student activity can also be used to check "similarity relations" among knowledge components; these relations are also useful for personalized recommendations. Knowledge components are typically organized in a hierarchy, and similarity relations can be naturally based on this hierarchy. However, it can be fruitful to supplement these with data based on student activity. This is analogous to the collaborative filtering approach used in recommender systems (Koren & Bell, 2015). Specifically, we have successfully used the "co-visitation" approach in an early version of the YouTube recommendation algorithm (Davidson et al., 2010): for a pair of knowledge components i and j, we compute the similarity $s_{ij} = \frac{c_{ij}}{c_i \cdot c_j}$, where c_{ij} is the number of students who solved both i and j, and c_i and c_j are counts of students for i and j.

3.5 Student-Item Interactions

To get a more detailed understanding of student activity within a system, it is useful to visualize student-item interactions. The basic form of this type of visualization is a matrix with students as rows, items as columns, and values in the matrix corresponding to some information about the interaction between a student and an item. The whole matrix is typically too large, and thus we need to use only a subset of students and items. Figure 4 shows examples.

[1] https://carrotsearch.com/foamtree/

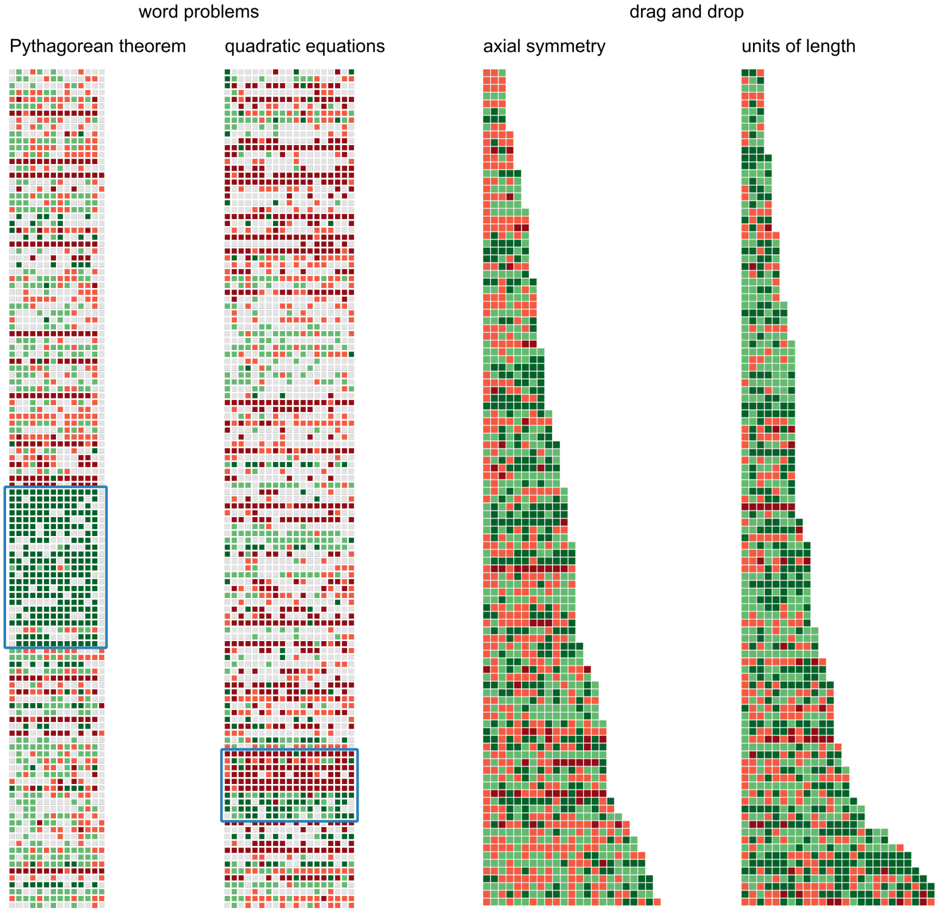

Figure 4. Student-item interaction visualization. Colour corresponds to performance (green = correct, red = incorrect; shade is based on response time). Blue rectangles denote suspicious student activity.

The general aim of this visualization is to provide insight into student behaviour and facilitate designers' decision making. Several specific examples are as follows:

- understanding the behaviour of users, personalization algorithms, and their interactions: the behaviour of people is messy and algorithms are adaptive; gaining insight into what is happening within the system is valuable;
- detecting and exploring undesirable behaviour by students, e.g., cheating, systematic guessing, or gaming the system: from the visualization, we can spot unexpected patterns and then build detectors to quantify them and find them systematically;
- getting inspiration for and insight into the development and evaluation of student models;
- understanding biases in data (discussed in more detail in the next section).

Grid visualization of student-item interactions can be done in many different ways, each suitable for different situations and purposes. The basic choices are as follows:

- Selection and ordering of students: To make the visualization usable, at most hundreds of students can be displayed, i.e., we need to use only a sample of students. The selection can be done as a random sample or a consecutive sample with respect to time (students within one week). We may want to filter only students satisfying certain conditions (e.g., sufficient activity). The ordering of students in the visualization can be according to timestamp (first activity), total activity, or success rate.
- Selection and ordering of items: A natural selection criterion for items is the unit in which they appear in the user interface, typically a knowledge component. For ordering, we may use their difficulty, their presentation order within the

system, or the timestamp of their creation. We may also use "per user" ordering, where items are sorted for each user (row) based on the activity of the given user (note that this approach leads to a different interpretation of columns).

- The values shown in the grid: Typically we want to visualize performance: correctness of answer or (normalized) response time. Additionally, we may want to visualize information about the ordering of actions or timestamps associated with actions. For easy readability, the values are best displayed by (discrete) colours, but it is also possible to use shades, sizes, or shapes.

Figure 4 shows two specific realizations for data from mathematics practice. In both cases, the colour corresponds to performance (correctness and response time). For word problems (left), students (rows) are sorted by their system IDs and problems (columns) are sorted by their difficulty. The shade is determined using a fixed threshold on response time (20 seconds). This visualization allows us to detect suspicious activities of students: blocks of students with consecutive IDs who have many fast, correct answers. For the drag-and-drop exercise (right), students are sorted by their activity, and problems within each row are sorted by the order in which a particular student solved them. The shades are based on relative response times (rank for a particular problem).

This visualization can help us estimate which type of student model and what kind of input information to use. Consider the case of the drag-and-drop exercise in Figure 4. It seems that Bayesian knowledge tracing, a commonly used student modelling approach, is not appropriate for this situation since it assumes a discrete transition from the unlearned to the learned state (Pelánek, 2017). Such a transition is not apparent in the figure. Also, since we can notice some consistency in response times, we may want to include them in a student model. In other cases, this visualization quite clearly shows different messages about the data. For English grammar exercises, the key information is the correctness of the answers, whereas response times are rather inconsistent. In programming exercises, correctness is not very informative since most attempts are eventually successful; the main source of data about student state is response time. The interaction matrix visualization cannot provide conclusive evidence for such claims. Nevertheless, it is a useful tool for guidance and prioritization of student modelling experiments.

4. Biases and Caveats

Data in learning systems contain many biases, patterns, and caveats. Student activity has strong temporal patterns: both regular (during the day, week, or year) and irregular (before final exams or during school closures due to the pandemic). Student behaviour is influenced by presentation aspects, e.g., ordering of items in the system. Some students are cheating or gaming the system. All of these issues influence the interpretation of data analysis; if we ignore them, we can reach misleading conclusions. On the other hand, data analysis can help us detect suspicious or unexpected trends, and these observations can lead to the improvement of a system.

4.1 Temporal Patterns

Data in learning systems contain strong temporal patterns. Figure 5 shows several examples. A basic pattern concerns student activity: there are strong patterns during the day, week, and year. Even within this basic activity pattern, there can be interesting differences. For example, some types of practice are more popular in the morning hours (use within school); others in the evening (practice at home). There are also changes in topic popularity; e.g., in our data on native language learning, basic orthography is particularly popular during autumn (the beginning of the academic year), whereas more complex syntax exercises are particularly popular during spring (the end of the academic year).

The temporal patterns concern not just activity but also performance. As Figure 5 shows, student success rate and guess rate[2] change significantly during weekends and holidays. This can be caused by various factors, e.g., changes in the student population (the sample of students who use a learning system on Saturday is different from those who use it on Tuesday) or changes in student setting and motivation (the same student has different conditions for answering a question during school hours on Tuesday and at home on Saturday).

As such, these observations are not surprising since they are a direct consequence of the ways learning is organized within the society (e.g., the organization of learning in schools, the order of topics in standard curricula). A detailed analysis of temporal patterns can suggest to system designers opportunities for their utilization, e.g., in the form of temporal recommendations (Koenigstein, Dror, & Koren, 2011).

More important, the presence of these patterns can have serious implications for research on and evaluation of results. For example, "black box" student models (like deep neural networks) may be able to achieve good predictive results by exploiting some of these temporal patterns and not by modelling learning. In the evaluation of student models, a common practice is to split data into training and testing sets, learn model parameters on the training sets, and report performance on the testing set.

[2] Guess rate is the ratio of answers that are wrong and very fast; i.e., the student answered the question without reading it properly.

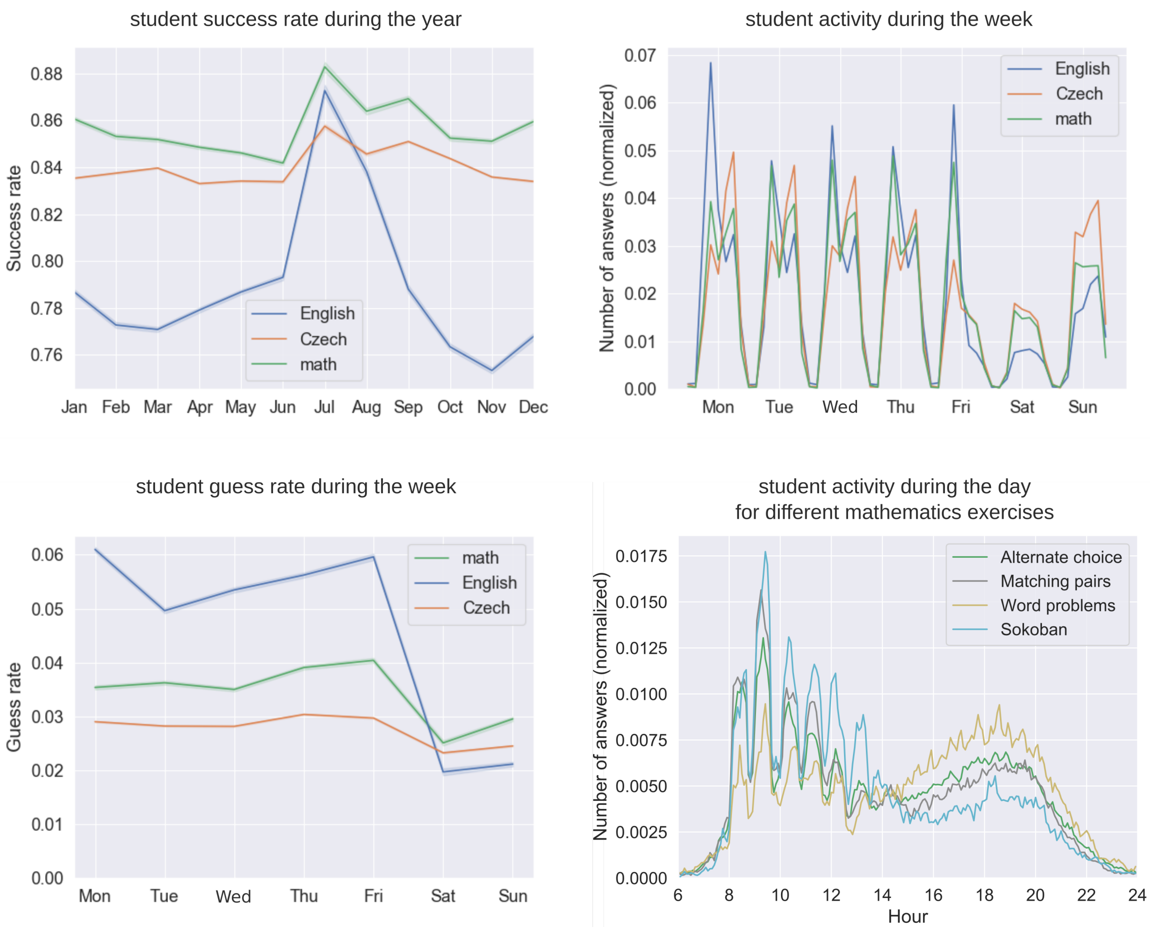

Figure 5. Illustrations of temporal patterns in data.

One way to split the data is by a time threshold. This can be a problem due to the presence of temporal patterns. Suppose that we split the data so the testing set contains only July and August data. These data are quite different from the data from the rest of the year—this could significantly influence the results of model comparison. This is a specific example of a general phenomenon called "dataset shift" (Moreno-Torres, Raeder, Alaiz-Rodríguez, Chawla, & Herrera, 2012).

4.2 Biases

The data in learning systems also contain biases. Specifically, items are not solved by a random sample of a student population but by a specific subsample. This is caused by several factors:

- Selection: Students choose (or are directed to by teachers) content that is adequate for them. This selection is influenced by external factors like a national curriculum.
- Presentation: The order in which items are presented (even when students are allowed to choose freely) influences the behaviour, typically as a preference for top items in a list—this is well documented, e.g., in search engine results (Joachims, Granka, Pan, Hembrooke, & Gay, 2017), and we see many instances of this preference in our learning data.
- System behaviour: If the system is adaptive and implements mastery learning or personalized recommendations, this can create specific biases and feedback loops, e.g., mastery attrition bias (Pelánek, 2018).

Some of these biases are quite obvious, e.g., it is quite clear that one-digit multiplication is solved by a different population than logarithmic equations. How to treat even these clear cases is not obvious. When is it sufficient to use simple statistics (like the success rate), which can be significantly influenced by these biases? When is it necessary to use more complex techniques (like item response theory models), which are able to take some of these biases into account? Which biases are most important for a particular system?

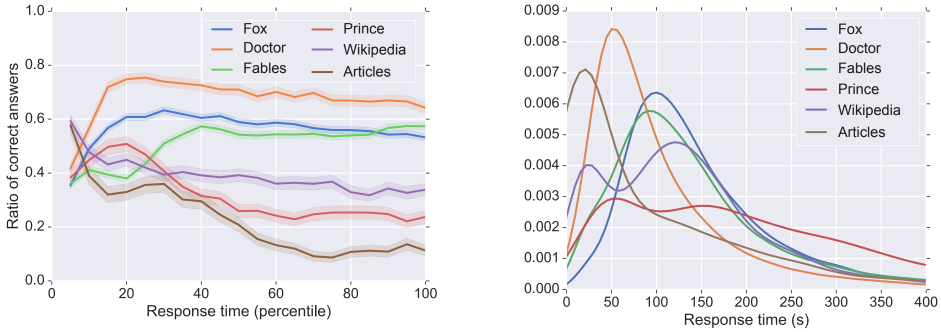

Figure 6. Analysis of data from a reading comprehension exercise. Each line corresponds to one set of items of similar complexity. Left: The relationship between speed (normalized response time) and correctness (the ratio of correct answers). Right: The distribution of response times (in seconds).

The presence of biases has important consequences for the use of datasets from learning systems in research, particularly in the evaluation of student models. To perform an evaluation, we need to make several methodological decisions, e.g., division of data into training and testing sets and the choice of a performance metric and the details of its computation (Pelánek, 2018). In order to choose an appropriate methodology, it is necessary to have a good understanding of the properties of a dataset and potential biases.

Unfortunately, some biases are quite nuanced, are hard to detect, or present themselves in a quite ad hoc fashion that is hard to incorporate into models. Consider the following specific examples, which are caused by ordering and presentation. In programming exercises, the first problem in each problem set has a lower success rate than others, even though it is objectively simpler than others (see Figure 1). This probably has several reasons, e.g., students learning the interface and students "just checking" the content of an item set. For logic puzzles, we see an analogous effect. However, there is an additional effect on the last item in the set (typically called "challenge"), which is attempted by many more students than other items in the relevant set. This is presumably caused by students who just want to see what a really difficult puzzle looks like but do not intend to try hard to solve it. Figure 6 (left) shows an analysis of the relationship between response time and success rate for several reading comprehension item sets. This relation differs significantly for different item sets. For two item sets (Fox, Doctor), we see an inverted-U relation, where very fast students have a low success rate. These two item sets are in the system presented on the top (as the first two, easiest sets). Part of the recorded activity (quick answers) thus probably corresponds to "exploring" behaviour, not serious attempts at solving. For two sets (Articles, Wikipedia), the highest ratio of correct answers is for the fastest responses, which is an indication of potential cheating.

4.3 Student Cheating

Whenever students have some external motivation to work in a learning system (e.g., homework, course credit), there is a risk of cheating. Here, we use the term *cheating* with a broad meaning—encompassing a range of activities where the collected data about student performance do not correspond to the real skill of students. This can be caused by a variety of reasons: answers submitted with the assistance of somebody else or directly by somebody else, use of multiple accounts to harvest correct answers (Northcutt, Ho, & Chuang, 2016), hacking the system (e.g., reading solutions from a page source code or JavaScript console), or gaming the system (exploiting the system's help and feedback features) (R. Baker et al., 2008).

Given the variety of cheating forms, it is typically impossible to avoid all types of cheating, and thus it is probable that some form of cheating is present in most learning systems. The extend of cheating, of course, depends on the type of system (e.g., consider a system for high-stakes testing versus a system for voluntary preparation). Even in systems where a low degree of cheating is expected, it is necessary to be aware of the possibility of cheating and to take it into account in the interpretation of the results of learning analytics. Even relatively small-scale cheating can lead to artificial patterns in data and misleading conclusions.

Specific forms of cheating can be detected by specialized detectors (Northcutt et al., 2016; Corrigan-Gibbs, Gupta, Northcutt, Cutrell, & Thies, 2015). Generic or unknown forms of cheating often manifest themselves by suspicious patterns in data. A specific example is provided in Figure 6—very low response times combined with a high success rate. The graph on the right shows the distribution of response times for the reading comprehension exercise. For the Articles and Wikipedia item sets, we see that a large portion of answers is suspiciously fast (these sets have texts that are longer and more complex than other sets). An investigation based on this analysis led to the realization that for this exercise, it was possible to determine correct answers from the HTML source code. This bug was exploited only by more senior students solving advanced item sets.

Here are some other examples of analyses that can lead to the detection of cheating:

- item difficulty and discrimination: low difficulty (compared to similar items), low or even negative discrimination index (i.e., performance on an item for weaker students is the same as or better than that for good students);
- artifacts in student-item interactions; Figure 4 shows an example of a math word problem—a temporal cluster of very fast correct answers, which may be due to copying of answers within a class;
- a sharp temporal change in difficulty or response time.

5. Discussion

We have provided an overview of analytics and visualization techniques for learning data. Whereas previous work mostly focuses on techniques targeted at students (Bull & Kay, 2007), teachers (Molenaar & Knoop-van Campen, 2017), or policymakers (Clow, 2012), we consider the perspective of a system designer. As we have discussed, for designers, analysis and visualization of learning data is a key tool with many applications, including determining development priorities, the management of meta-data, feedback for content authors, and impetuses for revisions.

The discussed techniques, however, are not only relevant for designers. Although we focused on the designer's perspective, there is clearly a significant overlap with techniques targeted at other users. For example, the visualization of student activity using treemaps is clearly also relevant for teachers. The designer's perspective can inspire the development of novel techniques targeted at teachers. For example, techniques for analyzing and visualizing item similarity (as illustrated in Figure 2) can be useful not just for system designers but also for teachers, e.g., for providing a quick overview of the content available in the system or as a tool for creating exams. Visualization of student-item interactions is also highly relevant for teachers.

The described techniques are also relevant for researchers who design and evaluate student models and personalization algorithms. Researchers should be acquainted with the properties of datasets that they use for their research; otherwise, they risk reaching misleading conclusions. As we have shown, data from learning systems contain many biases and caveats, and these can have a non-trivial impact on research results. The properties of real student data are often seen as an unpleasant hurdle for research. They can also, however, be seen as an opportunity that raises interesting research questions. How should we deal with biases in learning data? What techniques can be used to eliminate biases? What is a suitable methodology for evaluating student models under the presence of strong temporal patterns? How do different student models behave in the presence of cheating? Which models are most robust?

Declaration of Conflicting Interest

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The publication of this article received financial support from Faculty of Informatics, Masaryk University.

References

Aleven, V., McLaughlin, E. A., Glenn, R. A., & Koedinger, K. R. (2016). Instruction based on adaptive learning technologies. In R. E. Mayer & P. A. Alexander (Eds.), *Handbook of Research on Learning and Instruction.* New York: Routledge. https://doi.org/10.4324/9781315736419

Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. (2008). Why students engage in "gaming the system" behavior in interactive learning environments. *Journal of Interactive Learning Research*, *19*(2), 185–224. Retrieved from https://psycnet.apa.org/record/2008-07015-001

Baker, R. S. (2016). Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, *26*(2), 600–614. https://doi.org/10.1007/s40593-016-0105-0

Brennan, R. L. (1972). A generalized upper-lower item discrimination index. *Educational and Psychological Measurement*, *32*(2), 289–303. https://doi.org/10.1177/001316447203200206

Bull, S., & 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. Retrieved from https://dl.acm.org/doi/10.5555/1435369.1435371

Clow, D. (2012). The learning analytics cycle: Closing the loop effectively. In *Proceedings of the Second International Conference on Learning Analytics and Knowledge* (LAK 2012), 29 April–2 May 2012, Vancouver, BC, Canada (pp. 134–138). New York: ACM. https://doi.org/10.1145/2330601.2330636

Corrigan-Gibbs, H., Gupta, N., Northcutt, C., Cutrell, E., & Thies, W. (2015). Deterring cheating in online environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *22*(6), 1–23. https://doi.org/10.1145/2810239

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., ... Sampath, D. (2010). The YouTube video recommendation system. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (RecSys 2010), 26–30 September 2010, Barcelona, Spain (pp. 293–296). New York: ACM. https://doi.org/10.1145/1864708.1864770

De Ayala, R. J. (2013). *The Theory and Practice of Item Response Theory*. New York: Guilford Publications. https://doi.org/10.1007/s11336-010-9179-z

Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, *24*(4), 470–497. https://doi.org/10.1007/s40593-014-0024-x

Inventado, P. S., Inventado, S. G. F., Matsuda, N., Li, Y., Scupelli, P., Ostrow, K., ... McGuire, P. (2018). Using design patterns for math preservice teacher education. In *Proceedings of the 23rd European Conference on Pattern Languages of Programs* (EuroPLoP 2018), 4–8 July 2018, Irsee, Germany (pp. 1–8). New York: ACM. https://doi.org/10.1145/3282308.3282340

Joachims, T., Granka, L., Pan, B., Hembrooke, H., & Gay, G. (2017). Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 2005), 15–19 August 2005, Salvador, Brazil (Vol. 51, pp. 4–11). New York: ACM. https://doi.org/10.1145/1076034.1076063

Klinkenberg, S., Straatemeier, M., & 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. https://doi.org/10.1016/j.compedu.2011.02.003

Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2012). The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, *36*(5), 757–798. https://doi.org/10.1111/j.1551-6709.2012.01245.x

Koenigstein, N., Dror, G., & Koren, Y. (2011). Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (RecSys 2011), 23–27 October 2011, Chicago, IL, USA (pp. 165–172). New York: ACM. https://doi.org/10.1145/2043932.2043964

Koren, Y., & Bell, R. (2015). Advances in collaborative filtering. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 77–118). Springer. https://doi.org/10.1007/978-0-387-85820-3_5

Molenaar, I., & Knoop-van Campen, C. (2017). Teacher dashboards in practice: Usage and impact. In *Proceedings of the 2017 European Conference on Technology Enhanced Learning* (ECTEL 2017), 12–15 September 2017, Tallinn, Estonia (pp. 125–138). Springer. https://doi.org/10.1007/978-3-319-66610-5_10

Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, *45*(1), 521–530. https:/doi.org/10.1016/j.patcog.2011.06.019

Northcutt, C. G., Ho, A. D., & Chuang, I. L. (2016). Detecting and preventing “multiple-account” cheating in massive open online courses. *Computers & Education*, *100*, 71–80. https://doi.org/10.1016/j.compedu.2016.04.008

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. https://doi.org/10.1007/s11257-017-9193-2

Pelánek, R. (2018). The details matter: Methodological nuances in the evaluation of student models. *User Modeling and User-Adapted Interaction*, *28*(3), 207–235. https://doi.org/10.1007/s11257-018-9204-y

Pelánek, R. (2020a). Managing items and knowledge components: Domain modeling in practice. *Educational Technology Research and Development*, *68*(1), 529–550. https://doi.org/10.1007/s11423-019-09716-w

Pelánek, R. (2020b). Measuring similarity of educational items: An overview. *IEEE Transactions on Learning Technologies*, *13*(2), 354–366. https://doi.org/10.1109/TLT.2019.2896086

Pelánek, R., & Řihák, J. (2018). Analysis and design of mastery learning criteria. *New Review of Hypermedia and Multimedia*, *24*(3), 133–159. https://doi.org/10.1080/13614568.2018.1476596

Pelánek, R., & Řihák, J. (2016). Properties and applications of wrong answers in online educational systems. In *Proceedings of the Ninth International Conference on Educational Data Mining* (EDM 2016), 29 June–2 July 2016, Raleigh, NC, USA (pp. 466–471). Retrieved from https://www.educationaldatamining.org/EDM2016/proceedings/edm2016$_{s}hortpapers.pdf$

Ruipérez-Valiente, J. A., Muñoz-Merino, P. J., Leony, D., & Kloos, C. D. (2015). ALAS-KA: A learning analytics extension for better understanding the learning process in the Khan Academy platform. *Computers in Human Behavior*, *47*, 139–148. https://doi.org/10.1016/j.chb.2014.07.002

Van Der Linden, W. J. (2009). Conceptual issues in response-time modeling. *Journal of Educational Measurement*, *46*(3), 247–272. https://doi.org/10.1111/j.1745-3984.2009.00080.x

Wang, Y., Heffernan, N. T., & Heffernan, C. (2015). Towards better affect detectors: Effect of missing skills, class features and common wrong answers. In J. Baron, G. Lynch, N. Maziarz, P. Blikstein, A. Merceron, & G. Siemens (Eds.), *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge* (LAK 2015), 16–20 March 2015, Poughkeepsie, NY, USA (pp. 31–35). New York: ACM. https://doi.org/10.1145/2723576.2723618

Wauters, K., Desmet, P., & Van Den Noortgate, W. (2012). Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, *58*(4), 1183–1193. https://doi.org/10.1016/j.compedu.2011.11.020

Xhakaj, F., Aleven, V., & McLaren, B. M. (2017). Effects of a teacher dashboard for an intelligent tutoring system on teacher knowledge, lesson planning, lessons and student learning. In *Proceedings of the 2017 European Conference on Technology Enhanced Learning* (ECTEL 2017), 12–15 September 2017, Tallinn, Estonia (pp. 315–329). Springer. https://doi.org/10.1007/978-3-319-66610-5_23