Analysis of a Simple Model of Problem Solving Times*

Petr Jarušek and Radek Pelánek

Faculty of Informatics, Masaryk University Brno

Abstract. Our aim is to improve problem selection and recommendation in intelligent tutoring systems by modeling students problem solving times. We describe a simple model which assumes a linear relationship between latent problem solving ability and a logarithm of time to solve a problem. We show that this model is related to models from two different areas: the item response theory and collaborative filtering. Each of these areas provides inspiration for parameter estimation procedure and for possible extensions. The model is already applied in a widely used "Problem solving tutor"; using the data collected by this system we evaluate the model and analyse its parameter values.

Keywords: Problem solving, intelligent tutoring systems, item response theory, collaborative filtering.

1 Introduction

Problem solving is an important part of education in general and of intelligent tutoring systems in particular. To use problem solving activities efficiently, it is important to estimate well their difficulty – easy problems are boring, difficult problems are frustrating (this observation is elaborated by the flow concept [4]).

In intelligent tutoring systems [1, 12] problem selection is often done with respect to knowledge concepts – matching students mastery of concepts with concepts required to solve a problem. In some domains, however, there are many problems which are based on the same knowledge concepts, but differ significantly in their difficulty. In this work we focus on these types of domains, specifically on logic puzzles and introductory programming – these problems require little background knowledge, do not have easily identifiable skills, and yet span wide range of difficulty [6].

In this work we focus on predicting students problem solving times based on the data about previous problem solving attempts. To attain clear focus, we consider both students and problems as "black boxes", i.e., the only information that we use are the problem solving times. For practical application it may be useful to combine this approach with other data about students and problems (e.g., from knowledge tracing models [3]). Nevertheless, even the basic "black box" approach is applicable and has an important advantage of being simple

* This work is supported by GA ČR grant no. P202/10/0334.

and cheap (e.g., compared to knowledge tracing models which require significant expertise).

We describe a model which assumes a linear relation between a problem solving ability and a logarithm of time to solve a problem (i.e., exponential relation between ability and time). We provide connections of the model to two different areas – item response theory and collaborative filtering. Item response theory [2, 5] is used mainly in computerized adaptive testing to predict a probability of a correct answer and thus to select a suitable test item. Collaborative filtering [8] is used in recommender systems to predict user ratings of items (e.g., books) and recommend items to buy. Using inspiration from these areas we describe different variants of the model and different methods for parameter estimation.

The model is currently used in a "Problem solving tutor" – a web-based system which recommends students problems of suitable difficulty. The tutor contains more than 20 types of problems from areas of programming, math, and logic puzzles. The system is used in several schools and contains data about more than 5 000 users and 220 000 solved problems. Using this extensive data we evaluate the model and its different variants.

The evaluation shows several interesting results. The data support the basic model assumption of linear relation between ability and a logarithm of time to solve a problem. For predicting future times even a simple baseline predictor provides reasonable results; the model provides only slight improvement in predictions. Nevertheless, it brings several advantages. The model is group invariant and gives a better ordering of problems with respect to difficulty. It also brings additional insight – we can determine not just average difficulty of problems, but also their discrimination and randomness. With an extension of the model we can even determine similarity of individual problems (using just the problem solving times). All these parameters are useful for automatic problem selection in intelligent tutoring systems.

2 Modeling Problem Solving Times

We describe the setting, the basic models and we elaborate on its relation to the item response theory and collaborative filtering.

2.1 The Setting and Simple Models

We assume that we have a set of students S, a set of problems P, and data about problem solving times: t_{sp} is a logarithm of time it took student $s \in S$ to solve a problem $p \in P$ (i.e, t is a matrix with missing values). In this work we do not consider any other information about students and problems except for the problem solving times. We study models for predicting future problem solving times based on the available data. These predictions are denoted $\hat{t}_{sp}$.

As noted above, we work with a logarithm of time instead of the untransformed time itself. There are several good reasons to do so. At first, problem solving times have a natural "multiplicative" (not "additive") nature, e.g., if

Alice is a slightly better problem solver than Bob, then we expect her times to be 0.8 of Bob's times (not 20 second smaller than Bob's times). At second, previous research on response times in item response theory successfully used the assumption of log-normal distribution of response times [9, 11], analysis of our data also suggests that problem solving times are log-normally distributed. At third, the use of a logarithm of time has both theoretical advantages (e.g., applicability of simple linear models) and pragmatic advantages (e.g., reduction of effect of outliers).

Given our setting, the simplest way to predict problem solving times is to use mean time, i.e., $\hat{t}_{sp} = m_p$, where m_p is the mean of $t_{s'p}$ over students s' who solved the problem p. A straightforward way to improve and personalize this prediction is to take into account the performance of individual students. This leads to the "baseline" model $\hat{t}_{sp} = m_p - \delta_s$, where δ_s is a "mean performance of student s with respect to other solvers", i.e., $\delta_s = (\sum m_p - t_{sp})/n_s$, where n_s is the number of problems solved by the student.

Our basic model, on which we will further elaborate, is an extension of this baseline model. It is a linear model, which combines problem difficulty for average solver (b_p), problem discrimination (a_p) and student's ability (θ_s), i.e., $\hat{t}_{sp} = b_p + a_p\theta_s$. In the following we describe two different ways how to derive and further develop this basic idea.

2.2 Model Inspired by Item Response Theory

The item response theory (IRT) deals with test items with discrete set of answers and models the probability of a correct answer. There has been research on modeling response times in the context of IRT (see e.g., [9]), but in this research time is used only as an additional information (the main focus being on the correctness of response), not on the time itself.

The basic models of IRT assume that probability of correct response depends on one latent ability θ. The most often used model is the three parameter logistic model $P_{a,b,c,\theta} = c+(1-c)e^{a(\theta-b)}/(1+e^{a(\theta-b)})$. This model has three parameters (see Fig. 1): b is the basic difficulty of an item, a is the discrimination factor (slope of the curve, how well the item discriminates based on ability), and c is the pseudo-guessing parameter (lower limit of the curve, probability that even a student with very low ability will guess the correct answer).

In our setting, we similarly assume that a problem solving performance depends on one latent problem solving ability θ. We are interested in a "problem response function" $f(\theta)$, which for a given ability θ gives an estimate of a time to solve a problem. More specifically, the function gives a probabilistic density of times.

As a specific model we use the simplest "natural" choice: a normal distribution with the mean linearly dependent on the ability and with constant variance (remember that we work with a logarithm of time, i.e., this model assumes that the untransformed time to solve a problem is exponentially dependent on ability). The model thus has 3 problem parameters with the following intuitive meaning

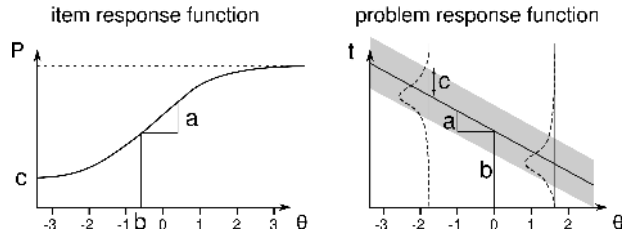

Fig. 1. An intuitive illustration of an item response function and a problem response function. Dashed lines illustrate distributions for certain abilities; solid line denotes the expected problem solving time, grey area depicts the area into which most attempts should fall. Note that we are dealing with a logarithm of time.

(we intentionally use notation analogical to IRT): discrimination factor a_p, basic difficulty of the problem b_p, and randomness factor c_p.

The problem response function, i.e., the probability that a student s with ability θ_s will solve a problem p at (a logarithm of) time t, is thus given by a normal distribution with a mean $b_p + a_p\theta$ and a variance c^2: $f_{a_p,b_p,c_p,\theta_s}(t) = \mathcal{N}(b_p + a_p\theta_s, c_p)(t)$.

The predicted time for a student s and a problem p is the expected value of f_{a_p,b_p,c_p,θ_s}, i.e., $\hat{t}_{sp} = b_p + a_p\theta_s$. The model and intuition behind its parameters are illustrated in Fig. 1. Discrimination factor a_p describes the slope of the function, i.e., it specifies how the problem distinguishes between students with different ability. Basic difficulty describes expected solving time of a student with average ability. The randomness factor describes variance in solving times for a particular ability.

Note that the presented model is not yet identified as it suffers from the "indeterminacy of the scale" issue in the same way as the basic IRT model. This is solved by normalization – we require that the mean of all θ_s is 0 and the mean of all a_p is -1.

Since we do not know either parameters of problems, or abilities of students, we need to estimate them from available data. Similarly to the procedures used in IRT [5], this estimation can be performed by iterative maximum likelihood estimation. We iteratively update problem (student) parameters assuming that student (problem) parameters are known. Maximum likelihood estimation for parameter values leads to ordinary least squares regression. Maximum likelihood estimation for students abilities gives the following update rule [7] (weighted sum of "local" ability estimates across all problems solved by a student): $\theta_s = \sum_p \frac{a_p^2}{c_p^2} \frac{t_{sp}-b_p}{a_p} / \sum_p \frac{a_p^2}{c_p^2}$.

2.3 Models Inspired by Collaborative Filtering

Collaborative filtering is a method used in recommender systems, e.g., systems for recommending movies (Netflix) or books (Amazon). The goal in these cases is to predict future user ratings based on past ratings. Instead of predicting ratings,

we predict problem solving times, but otherwise our situation is analogical (in both cases the input is a large sparse matrix).

There are two basic methods for collaborative filtering: neighbourhood based (memory based) and matrix factorization (model based). The main principle of matrix factorization methods is based on singular value decomposition (SVD) – a linear algebra theorem which states that any matrix A can be decomposed as $A = UDV^T$, where D is a diagonal matrix and U, V are orthonormal matricies. Using this decomposition it is possible to find an approximation of A by using only first few rows in the product.

This theorem can be directly used only for complete matrices. In collaborative filtering, however, there are typically many missing values. This can be overcome by imputing data (e.g., substituing means in place of missing values), but such approach has many disadvantages (e.g., imprecision, computational demands). It is preferable to construct directly an approximation of the form: $\hat{r}_{ij} = \boldsymbol{p_i}^T \cdot \boldsymbol{q_j}$, where $\hat{r}_{ij}$ is the predicted rating and $\boldsymbol{p_i}$ and $\boldsymbol{q_j}$ are feature vectors of length k. This model is typically further extended to include the baseline prediction for a given item [8]. This leads (using our notation) to the following model: $\hat{t}_{sp} = b_p + \boldsymbol{a_p}^T \cdot \boldsymbol{\theta_s}$, where $\boldsymbol{a_p}$ and $\boldsymbol{\theta_s}$ are vectors of length k which specify problem-feature and user-feature interactions. The parameters of the model are typically estimated using stochastic gradient descent with the goal to minimize sum of square errors [8].

Note that for $k = 1$ the resulting model has the same structure as the model inspired by IRT. Both the item response theory model and our analogical model of problem solving times can also be extended to incorporate multidimensional ability [10].

Collaborative filtering has to deal with parameter changes during time (e.g., user book preferences evolve) [8]. Similarly, in our setting it is sensible to incorporate learning into the model – students problem solving ability should improve as they solve more problems. A natural extension of the model is the following: $\hat{t}_{sp} = b_p + a_p(\theta_s + \Delta_s \cdot f(k))$, where k is the order of the problem in problem solving sequence; f is a monotone function, and Δ_s is a student's learning rate.

2.4 Group Invariance

The mean predictor and the simple baseline predictor are misleading if the subgroup of students which solved a particular problem is not representative of the whole population. An important feature of our approach is that the models are "group invariant" (similarly to IRT models [5]), i.e., problem (student) parameters do not depend on a subgroup of students which solved the problem (problems solved by a student).

Let us describe this important feature on a specific example. When we have a set of problems, then typically the harder problems are solved only by students with above average ability. If we use a mean problem solving time as a predictor of problem difficulty, than we underestimate the difficulty of these harder problems. Our model takes abilities of solvers into account and thus the obtained problem parameters are independent of the group of solvers.

3 Application and Evaluation

Now we briefly introduce a "Problem solving tutor", which uses the described approach to make predictions and recommendations to students. Data collected by this system are used for evaluation of described models.

3.1 Problem Solving Tutor

The described approach is currently used in a "Problem solving tutor" – a free web-based tutoring system for practicing problem solving, which is available at `tutor.fi.muni.cz`. At the moment the system focuses solely on the "outer loop" of intelligent tutoring [12], i.e., recommending problem instances of the right difficulty.

The system contains more than 20 types of problems, particularly computer science problems (e.g., binary numbers, robot programming, turtle graphics, introductory C and Python programming), math problems (e.g., describing functions, matching expressions), and logic puzzles (e.g., Sokoban, Nurikabe). All problems are "pure" problem solving problems with clearly defined correct solution – problem solving time is the single measure of students performance, there are no "quality of solution" measures (i.e., no hints during solutions or acceptance of partial solutions).

The system was launched in March 2011, it is already used by more than 20 schools and has more than 5 000 registered users (mainly university and high school students) who have spent more then 8 000 hours solving more than 220 000 problems. The collected data are used for the below described evaluation. The number of solved problems is distributed unevenly among different problem types, in the evaluation we use only problems for which we have sufficient data.

3.2 Analysis of Parameter Values

We begin our evaluation by analysis of the basic model with one ability. The parameter values were estimated as described in Section 2. We have described two ways to derive our basic model and estimate parameters: model inspired by item response theory with parameters estimated by alternating maximum likelihood estimation and the SVD inspired model with parameters estimated by stochastic gradient descent (the specific algorithm parameters were used as in [8]). Our results show that these two ways to estimate the parameters lead to nearly the same results. Thus here we report only on the computed parameters (student abilities θ, problem parameters a, b, c) of the IRT inspired model.

Student abilities should be normally distributed in a population, and the results show that the estimated abilities θ are really approximately normally distributed (see Fig. 2.). The variance of the distribution depends on the problem type – for educational problems we have larger variance of abilities than for logic puzzles.

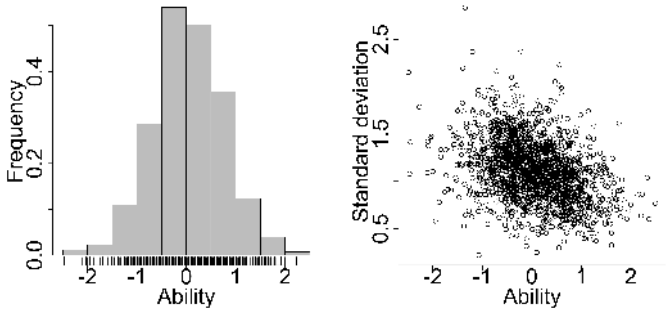

Fig. 2. Left: Distribution of abilities for the Robotanist problem. Right: Ability versus variation in the student performance for the Robotanist problem.

Generally the data suggest that the basic assumptions on which the model is based are suitable, e.g., for particular problems the relation between the estimated ability θ and a logarithm of time is really linear as the model assumes. Nevertheless, one result shows that some of the model assumptions are too simple. Fig. 2. shows a relation between an estimated ability and a variation in student performance (the standard deviation of ability estimates for individual problem instances). There is a slight negative correlation, i.e., students with lower ability have larger variance, whereas the model assumes a constant variance. Thus the model can be extended by another parameter to describe this decrease of variance with increasing ability.

Fig. 3. shows scatter plots for problem parameters a, b, c. There is a correlation between the basic problem difficulty and its discrimination – more difficult problems are more discriminating. The randomness parameter (which corresponds to variance of problem solving times) is nearly independent of the basic problem difficulty (there is a positive correlation, but only small). Note that this result indirectly supports the application of logarithmic transformation of times. If we had used untransformed times or some different transformation, there would be much stronger dependence.

Although there are some correlations among the parameters, generally the parameters are rather independent, i.e., each of them provides a useful information about the problem difficulty. For example, in intelligent tutoring system, it may be suitable to filter out problems with large randomness or low discrimination.

3.3 Evaluation of Predictions

Evaluation of model predictions was done by repeated random subsample cross-validation. We performed 20 repetitions, each with 90% of data as a training set and the remaining 10% of data as a test set. Table 1. compares the results using the root mean square error metric. We have also evaluated other metrics

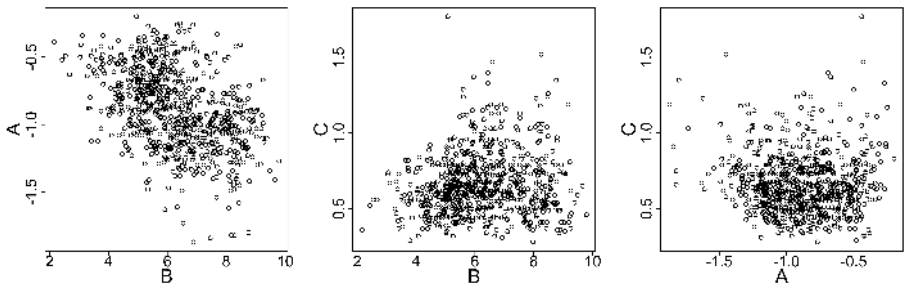

Fig. 3. Relations between parameters a, b, c of the model.

Table 1. Quality of predictions for different models and problems measured by root mean square error metric. Baseline model, IRT-model, and SVD-model are models described respectively in Sections 2.1, 2.2, 2.3. All models assume a single latent ability.

Problem type	Mean time	Baseline model	IRT-model	SVD-model
Binary numbers	1.1717	0.9941	0.9856	0.9860
Graphs and functions	1.2868	1.0477	1.0395	1.0419
Nurikabe	0.9021	0.7111	0.7191	0.7175
Robotanist	1.3137	1.2056	1.1944	1.1963
Rush hour	0.8937	0.8072	0.7948	0.7975
Slither Link	1.0252	0.7873	0.7766	0.7760
Sokoban	1.1491	0.8965	0.8876	0.8893
Tents	1.0238	0.9355	0.9423	0.9434
Tilt maze	1.0044	0.8665	0.8620	0.8656

like the Pearson and Spearman correlation coefficients and mean absolute error. The relative results are very similar.

The results show that all models provide improvement over the use of a mean time as a predictor. Most of the improvement in prediction is captured by the baseline model; models with more parameters bring a slight, but not very important improvement. As mentioned above, IRT-based and SVD-based parameter estimations lead to nearly the same parameter values and thus the predictions are also nearly the same.

So far we have evaluated absolute predictions of problem solving time. In practical applications it may be more important to focus on relative predictions, i.e., on ordering of individual problem instances, so that students can progress from easy problems to difficult ones. Here the group invariance issue (described in Section 2.4) becomes important. The baseline model leads to same ordering of problems as the mean time, i.e., it is not group invariant, whereas other described models are group invariant. An analysis of data shows that the ordering based on our models is better than the ordering based on mean time (to make this comparison we ordered problems into sequences $P_1, \ldots, P_n$ and counted how many times did some student solve problem P_i faster than problem P_j for $i > j$).

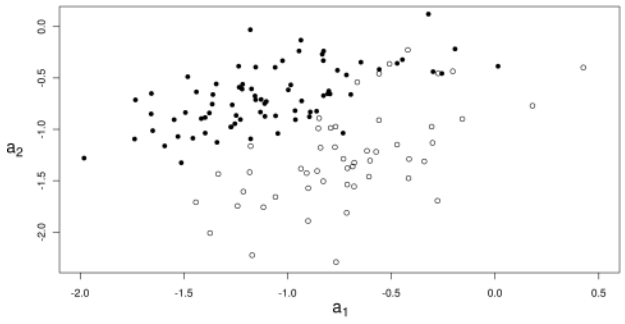

Fig. 4. Determination of problem similarity by the extended model with two abilities. Graph axes are the problem discrimination parameters a_1, a_2. White dots are Sokoban problems, black dots are Slither Link problems.

3.4 Extended Models

Finally, we provide a brief evaluation of extended models described in Section 2.3 – a model with learning and a model which assumes multiple abilities (i.e., model corresponding to SVD technique with several features). Parameters for these models were estimated using the stochastic gradient descent method in a similar way as for the basic model.

On our current data these models do not improve predictions due to the overfitting – we get improved fit over the training set, but worse fit on the test set. Nevertheless, even with the current data these models can give us some interesting insight.

The model with learning is of the following form: $\hat{t}_{sp} = b_p + a_p(\theta_s + \Delta_s \cdot f(k))$, where k is the order of the problem in a problem sequence, f is a monotone function, and Δ_s is a learning rate. Our analysis confirms an intuitive expectation that f should be sublinear (learning is faster at the beginning and then slows down); a use of square root leads to a good fit. Results also show that for our problems the learning rate Δ_s is weakly positively correlated with ability θ_s (i.e., better students improve faster).

We also evaluated a model with two abilities: $\hat{t}_{sp} = b_p + a_{1p}\theta_{1s} + a_{2p}\theta_{2s}$. Although the model does not improve predictions on our current data due to the overfitting, we can at least evaluate whether the automatically learnt concepts (abilities) are sensible. To do so we performed the following experiment: we mix data for two types of logic puzzles, let the algorithm learn the concepts, and then check, how well are the puzzles separated. Fig. 4 shows results for two particular problems. As we can see, the two problem types are separated quite well by the automatically learnt concepts. This extended model can thus be used for automatic determination of similarity between problems within a given set of problems. This can be useful for problem recommendation in intelligent tutoring

systems. If a student solved a particular problem slowly, we can give her a similar problem, but easier problem, if a student solved problem quickly, we can give her a problem utilizing different concept.

4 Conclusions

We describe a model of students problem solving times, which assumes a linear relationship between a problem solving ability and a logarithm of time. We derive the model details and parameter estimation procedures from two different areas: the item response theory and collaborative filtering. The model is already applied in an online "Problem solving tutor" to recommend problems of suitable difficulty. This system is already widely used (more than 220 000 problems solved), the collected data were used for evaluation of the model. The results show that the model brings only slight improvement compared to the baseline predictor, but also that the model provides interesting information about problems (including determination of problem similarity based only on problem solving times). This information can be useful for problem selection and recommendation in intelligent tutoring systems.

References

1. Anderson, J., Boyle, C., Reiser, B.: Intelligent tutoring systems. Science 228(4698), 456–462 (1985)
2. Baker, F.: The basics of item response theory. University of Wisconsin (2001)
3. Corbett, A., Anderson, J.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction 4(4), 253–278 (1994)
4. Csikszentmihalyi, M.: Flow: The psychology of optimal experience. HarperPerennial New York (1991)
5. De Ayala, R.: The theory and practice of item response theory. The Guilford Press (2008)
6. Jarušek, P., Pelánek, R.: What determines difficulty of transport puzzles? In: Proc. of Florida Artificial Intelligence Research Society Conference. pp. 428–433. AAAI Press (2011)
7. Jarušek, P., Pelánek, R.: Modeling and predicting students problem solving times. In: Proc. of International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2012). LNCS, vol. 7147, pp. 637–648. Springer (2012)
8. Koren, Y., Bell, R.: Advances in collaborative filtering. Recommender Systems Handbook pp. 145–186 (2011)
9. Van der Linden, W.: A lognormal model for response times on test items. Journal of Educational and Behavioral Statistics 31(2), 181 (2006)
10. Mulder, J., Linden, W.: Multidimensional adaptive testing with kullback–leibler information item selection. Elements of adaptive testing pp. 77–101 (2010)
11. Van Der Linden, W.: Conceptual issues in response-time modeling. Journal of Educational Measurement 46(3), 247–272 (2009)
12. Vanlehn, K.: The behavior of tutoring systems. International Journal of Artificial Intelligence in Education 16(3), 227–265 (2006)