Behavioural Cloning of Teachers for Automatic Homework Selection

Russell Moore, Andrew Caines, Andrew Rice, and Paula Buttery

ALTA Institute, Department of Computer Science & Technology
University of Cambridge
15 J. J. Thomson Avenue
Cambridge, U.K.
{rjm49,apc38,acr31,pjb48}@cam.ac.uk
http://alta.cambridgeenglish.org

Abstract. We describe a machine-learning system for supporting teachers through the selection of homework assignments. Our system uses behavioural cloning of teacher activity to generate personalised homework assignments for students. Classroom use is then supported through additional mechanisms to combine these predictions into group assignments. We train and evaluate our system against 50,065 homework assignments collected over two years by the *Isaac Physics* platform. We use baseline policies incorporating expert curriculum knowledge for evaluation and find that our technique improves on the strongest baseline policy by 18.5% in Year 1 and by 13.3% in Year 2.

Keywords: homework selection · behavioural cloning · deep learning

1 Introduction

Tutoring by human teachers is known to produce large learning gains for students, whether one-to-one or in groups [5]. For larger groups, a teacher's attention must inevitably be spread more thinly across their students. We seek to develop learning technologies which can support teachers in this scenario.

We focus on setting homework assignments. The setting of good-quality homework is of recognised pedagogical importance [14, 30], but how work should be chosen is less clear: human teachers have been shown to adopt varied strategies of teaching even for very simple lessons [16]. In light of this we developed a system, `HWGen` that learns from expert actions using behavioural cloning.

`HWGen` generates homework assignments at both the individual and group level which we compare to assignments derived from a selection of naive and curriculum-aware baseline policies.

We trained our system with two years of student data from *Isaac Physics*, a major online teaching platform and find that our model is able to suggest assignments that are closely aligned with the choices made by human teachers. At the individual level, `HWGen` improves on the strongest baseline policy by 17.8% in Year 1 and by 12.9% in Year 2. Making homework selections per group, `HWGen` improves on the strongest baseline policy by 18.5% in Year 1 and by 13.3% in Year 2. These results reflect the way that human teachers make group-level homework decisions based on class progress as a whole, but also that setting personalised assignments stays fairly close to teacher decisions while also taking into account individual factors.

In this paper we make the following contributions: 1) we give a neural-network architecture for suggesting personalised homework assignments; 2) we propose mechanisms to combine personalised predictions in order to generate group assignments; 3) we use data on student and teacher behaviour from the *Isaac Physics* platform to show that our technique significantly outperforms a variety of baseline policies.

2 Related Work

Personalisation in educational technology is an exciting prospect since learners are known to progress at different rates and in different styles [28, 4, 8] and personalised tutoring has been shown to have beneficial effects on student learning [27]. Lindsey and colleagues found that a personalised review system for course content yielded a 16.5% boost in retention rates over standard practice (massed study) and a 10% improvement over a one-size-fits-all strategy for spaced study [18]. Advancing personalised learning is recognised as one of the National Academy of Engineering's *Grand Challenges* for the 21$^{\text{st}}$ Century.[1] Such systems traditionally involve considerable effort and pedagogical knowledge to design, author, and structure content [20, 33, 10]. One example is the use of concept maps [22] to represent the structure of skills and knowledge: these can help guide learning [2] and are used in tutoring systems [12] but they require expert knowledge and careful design.

There has been work on intelligent tutoring systems to imitate teachers and widen educational access since the 1960's [34], but successful autonomous selection of the right homework task at the right time remains elusive. Rather than developing an analytical solution, we hope instead to learn from the usage data logged by education platforms.

Teaching data combined with student data provide an appropriate setting for *imitation learning*. In imitation learning, a system learns appropriate responses to its environment from a human actor. The archetypal scenario is the self-driving car (in which the vehicle controller learns from a human driver) [25], but it has broader applications, and it has recently been used to provide personalised navigation for web-based learners [23].

In particular *behavioural cloning* is a simple but widely used form of imitation learning [6], useful in situations where non-interactive data is available for training, where actions are judged immediately (*i.e.* no long term reward information is available), and where imperfect actions are unlikely to lead to cascading errors. In a classroom setting there is no risk of physical mishap, and

[1] `http://www.engineeringchallenges.org/challenges/learning.aspx`

teachers are able to exercise expert judgement with respect to any suggestions a system might make: this makes behavioural cloning a viable candidate as an approach.

3 Experiments

This work is based on user data from *Isaac Physics*,[2] a UK government funded project aimed at pre-university students. Launched in 2015, *Isaac* serves physics and mathematics exercises to over 120,000 registered users globally. As well as for private study, the platform is designed for use by school classes with homework assignments set by their teachers. It is the actions of these teachers that we aim to imitate.

There are two related sub-tasks in this work: (1) the task of setting homework for an individual, and (2) of setting homework for a group. The homework selection should be available 'on demand', so that a teacher setting work for a class could use the software as a teaching aid, or so that a student can obtain personalised suggestions for independent study.

3.1 Data

This work uses a dataset taken from *Isaac* activity logs collected from March 2015 to March 2018. These are general-purpose event logs which record when users visit a page, answer a question (correctly or otherwise), view a hint, and so on. The logs also track when teachers assign homework to their class (a *SET_ASSIGNMENT* actions) and it is this behaviour we aim to clone.

Questions on Isaac are organised into *pages*: a page being the smallest teaching unit which may be assigned as homework. Students are organised into *groups*, which can be thought of as virtual classes. A teacher's *SET_ASSIGNMENT* action points a group of students to one or more question pages.

The *Isaac* project publishes a number of textbooks that accompany the platform[3]. Of these, the earliest book *Mastering Essential Pre-University Physics* ('the textbook'), currently accounts for around 80% of the homework set on the platform, with material across 73 pages forming the curriculum from which `HWGen` selects work. There are hundreds more question pages in *Isaac* but they are not all ordered for difficulty. By using only the textbook questions we have a controlled experiment in which the questions have an implicit canonical ordering – the order of page numbers from 1 to 73 – which we can use for baseline selection policies. If `HWGen` proves to be successful at this task, it can then be extended to the wider unordered curriculum.

Data was filtered so that only students between 16 and 18 years (the target age for the textbook) where considered. We wanted to imitate teachers who are engaged and who know their students well – to this end we removed teachers who

[2] `https://isaacphysics.org`

[3] `https://www.isaacbooks.org`

rarely set work (< 5 assignments) and excluded large groups ($\geq$ 30 students). We ranked teachers by activity, choosing the most prolific first. We measure each student's time on the platform from the date of their first assignment, and we split the students into a Year 1 group and Year 2 group at 365 days of use.

After filtering, 6672 instances of group-level assignments (50,065 assignments to individual students) were available for training and validation. For testing, 970 group-level assignments (6028 individual assignments) were chosen from a separate held-out set of teachers. A box-and-whisker plot of the training data is given in Figure 1, showing a noisy upward trend between students' time on the platform in days and the position of homework assignments in the textbook in pages. From this we can see that although there is evidence of the curriculum being followed, we should not expect a linear predictor to perform well due to noise, especially in the tail of the data where homework level becomes less consistent.

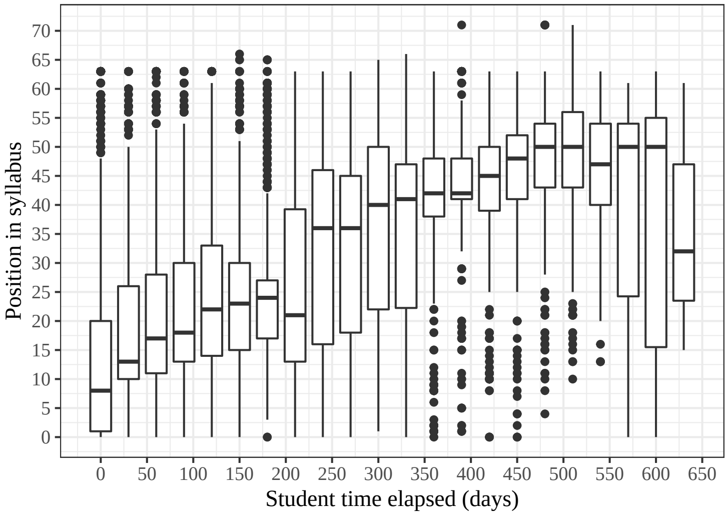

Fig. 1. The training dataset summarised in two dimensions: student duration on platform (days) is on the x-axis, position in textbook of homework selected (page number) is on the y-axis. The boxes show the range between first and third quartiles, the horizontal stripe indicates the median, the whiskers represent $1.5 * IQR$ where IQR is the inter-quartile range (third minus first quartile); other points beyond the whiskers are outliers. There are 12,489 features in the full dataset.

3.2 Student features

The student is represented as a combination of four vectors: $\boldsymbol{x}$ for practice, $\boldsymbol{u}$ for success, $\boldsymbol{a}$ for previously assigned work, and a small set of real-valued features

$\boldsymbol{s}$. There are 2605 elements in $\boldsymbol{x}$ and $\boldsymbol{a}$, one for each page on *Isaac*, and 7274 in $\boldsymbol{u}$ (for each page sub-part) and these take binary values depending on whether a page is attempted/assigned, or a sub-part is completed. Note that although we only set homework from the textbook, we track students' encounters with all pages on the platform: many of the pages outside the book are more difficult (or more involved) and so a student who has attempted such questions may be ready for more advanced material.

The $\boldsymbol{s}$ vector contains the following statistics about the student: *age* (in years, rounded to 1 decimal place), *days on platform*, *number of distinct questions attempted*, *number of attempts at all questions*, *number of questions passed.*

Other features were measured but not included in the model as they provided no benefit to performance, including: questions passed per day, question pass-rates, and the number of attempts the student had taken at each question (effectively this was clipped to binary in the $\boldsymbol{x}$ vector). Finer grained breakdowns of attempts, and 'recent history' versions of these features also made no improvement. Several of the profile features were tried in combination (e.g. question attempts per day as a proxy for motivation) but these were less successful than allowing the neural network to learn its own internal representations.

3.3 Implementation

The homework-selection task was formulated as a multiclass classification problem, using a feed-forward neural network (Figure 2). Hidden layers are all rectified linear units (ReLU), except for a concatenation layer to merge encodings. These are followed by two more ReLU layers each suffixed by a dropout layer ($rate = 0.2$). Layer structure, dropout, learning rate and batch size, were found by randomised parameter search.

The network output is passed through a softmax activation function (1) to convert internal scores into a probability distribution. The function transforms the j^{th} element of the raw neural network output, $\boldsymbol{z}$, into a probability conditioned on the input ($\boldsymbol{s}$,$\boldsymbol{x}$,$\boldsymbol{u}, \boldsymbol{a}$), $Pr(y = j \mid (\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{a}))$. The output of the network consists of 73 units, one for each possible homework choice.

$$softmax(\boldsymbol{z})_j = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)} \tag{1}$$

The software used in this work was implemented in `Python 3.6` using `Keras` [11] with a `Tensorflow` [1] back-end, and `scikit-learn` [24] for the machine learning components.

The network was trained with a categorical cross-entropy loss function using the Adam optimiser [17] with best results at $\alpha = 0.001$, $batch = 32$. Real-valued inputs to the network were centred and scaled to have unit variance and zero mean. Binary inputs were not transformed.

From the 50,065 samples in the training data, 10,013 (20%) were used for validation and to trigger early-stopping.

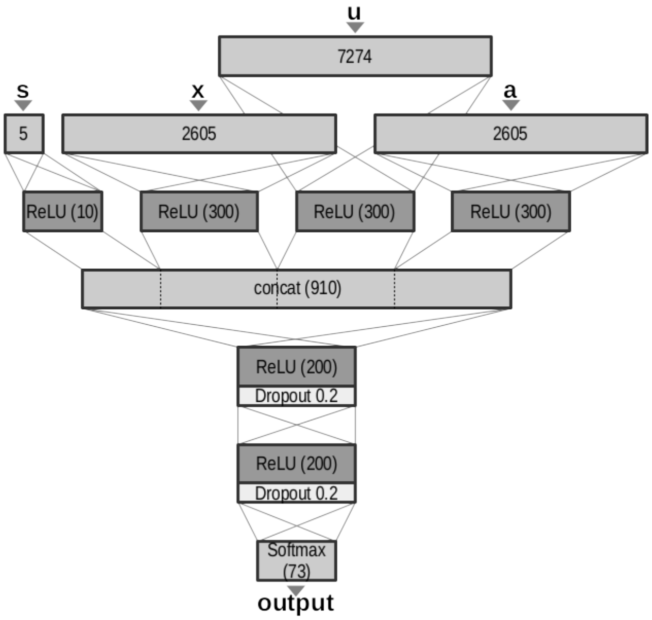

Fig. 2. Outline of the neural architecture. Input layers are at the top. Cross-hatched areas denote densely connected layers.

3.4 Setting Individual Work

The homework-setting decision is treated as an action-selection task, modelled as multiclass classification. The neural network outputs a softmax vector, the j^{th} element of which is the probability of taking the j^{th} action as determined from training. We choose the action given the highest probability by the network.

3.5 Modifications to Set Group Work

For setting homework at the group level, two alternative methods were applied to select a page for the group's next assignment:

- *Softmax averaging*: the softmax vectors for all students in the group are summed, and normalised by the group size. This produces a new softmax vector, and the top-scoring element is selected.
- *Voting*: the top-scoring candidates are chosen from each student's softmax vector. These are then counted, and the candidate that receives the most 'votes' is selected.

Both approaches can be implemented with simple control logic.

3.6 Baseline Policies

The performance of HWGen (in both individual and group modes) was judged against three baseline policies: a random policy and two *oracle* policies. The random policy randomly chooses an assignment from those that the student has not yet attempted.

The two oracle policies are 'curriculum aware'—they have access to ordering knowledge about the material that is not available to HWGen. The ordering was developed by the authors of the textbook as the logical progression of content.

The *linear* oracle policy selects the material in book order at isochronal intervals; the *step* oracle policy selects the next item from the book that has not yet been attempted by the student.

3.7 Evaluation

Our investigation into the Isaac data (Figure 1) showed that teachers adhere to the curriculum order in the textbook (as captured by the oracle policies) but only on average. We therefore hypothesise that HWGen should outperform these policies by capturing the context of each assignment choice.

We are interested to know how closely a homework selection policy will match a human teacher in a given situation. Using the textbook page numbers to give an ordering, the policy is a map $\pi : (\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{a}) \rightarrow \hat{y}$ from the student's vectors onto the index of the page of the assignment to be set. The true index, y, giving the choice actually made by the human teacher in this situation, is not known to the policies and is only discovered during testing.

Each $\hat{y}$ is evaluated against the real selection y, by taking $(\hat{y} - y)$. This gives an integer score in the range [-72,72], where zero is the target. Having values in a range allows partial credit to be assigned. Note that a negative or positive score indicates a policy that lags or leads the human teacher, respectively.

The above steps are repeated for every $((\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{a}), y)$ pairing in the test data. The policy is summarised by a standard metric, root mean squared error, which provides interpretable performance measures in terms of number of pages deviated (see Figure 3).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

Fig. 3. Root mean squared error (RMSE) of a predicted set of target values, $\{\hat{y}\}_1^N$, from their true values, $\{y\}_1^N$.

4 Results

Results from the experiment are presented in Table 1. In Year 1, HWGen with group voting has the highest accuracy in terms of the proportion of matches with teacher selections. The other HWGen policies perform similarly on this measure, as does oracle-step. However, oracle-step is on average slightly ahead of the teacher's point in the textbook (a positive lead/lag value) whereas the HWGen policies all lag slightly behind (negative values). However, all HWGen policies have lower RMSE than oracle-step, indicative of a tighter fit around the gold-standard teacher selections.

Table 1. Accuracy (prop. exact matches), mean lead (+) or lag (-) compared to teacher selections, and root mean squared error for HWGen and baseline homework-setting policies (*significantly different to best baseline, $p < 0.001$; †significantly different to HWGen for individuals, $p < 0.001$).

	Year 1 (n=6111)			Year 2 (n=572)		
Policy	Acc.	Lead/Lag	RMSE	Acc.	Lead/Lag	RMSE
Human teacher (target)	1	0	0	1	0	0
Random	.015	+17.2	31.4	.009	**+.091**	25.4
Oracle linear	.095	-6.24	20.2	0	+42.3	48.6
Oracle step	.137	**+1.18**	17.4	**.094**	+2.49	12.5
HWGen	.132	-2.37	14.3*	.017	-4.98	10.9*
HWGen group vote	**.141**	-2.81	14.4†	.019	-5.06	**10.8**†
HWGen group softmax ave.	.133	-2.82	**14.2**†	.014	-5.51	11.2

The same is true for Year 2, though in this case the policy with the highest proportion of exact matches is oracle-step, likely because as the choice of outstanding items in the curriculum narrows, a policy which draws from those remaining items in sequence will tend towards the ground truth. Note that HWGen is allowed to select from all textbook pages and so does not have this advantage of a narrowing pool. Being able to re-select items as revision work is a desirable mechanism however. Again, we see that oracle-step stays slightly ahead of the teacher while HWGen lags behind. The random baseline has the best lead/lag score (closest to zero), but this masks the high variance in its choices, shown by an RMSE much higher than HWGen.

In both Year 1 and Year 2, the lowest RMSE (our primary evaluation metric) is one of the HWGen group policies – softmax averaging in the first year, voting in the second year. This is expected, as we train on group level decisions made by the teachers. What is encouraging though is that HWGen for individuals outperforms all baseline policies too, indicating that it will serve individual users of intelligent tutoring systems as well as class groups.

In Figure 4 we show a density plot of the different spreads of HWGen and the baseline policies in terms of number of pages difference from the teacher's choices (x=0). We see that in Year 1 HWGen very closely mimics teacher behaviour, with

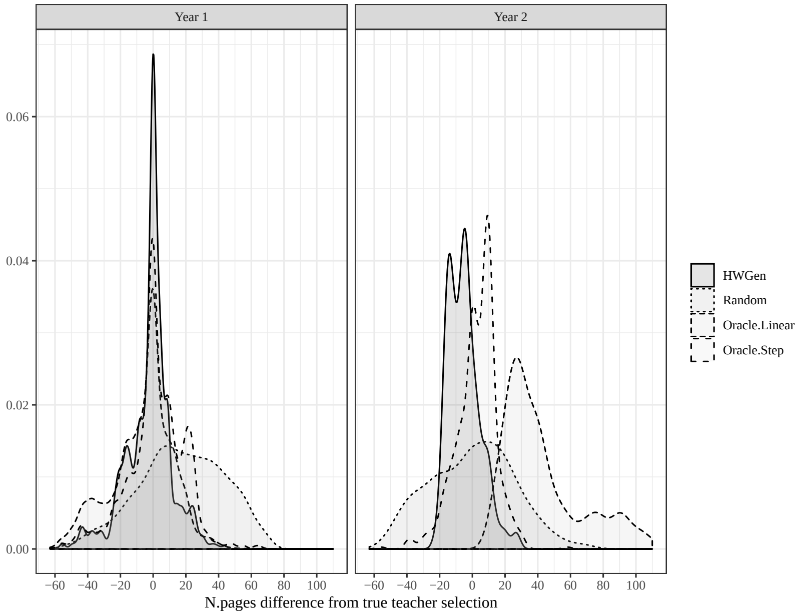

Fig. 4. Density plot of differences for each homework selection by HWGen and baselines: difference from the true selection on the x-axis (n.pages), density on the y-axis (sums to 1 for each curve).

the tallest peak around zero on the x-axis, while in Year 2 it still clusters around ground-truth but much of its decisions lag behind zero whereas oracle step is also grouped close to zero but in contrast tends to lead the teacher by a few pages. It may be that pedagogically it is preferable to move through a curriculum more slowly than more rapidly, and therefore we might prefer `HWGen`'s more conservative decisions.

We find `HWGen` and oracle step homework differences ($true.homework - predicted.homework$) are statistically separable by paired sample t-tests both for Year 1 ($t = 21.4, df = 6110, p < 2.2e\text{-}16$) and Year 2 ($t = 14.8, df = 571, p < 2.2e\text{-}16$). Meanwhile the same is true of `HWGen` versus `HWGen` group vote in ($t = 5.79, df = 6110, p = 7.6e\text{-}09$) and `HWGen` group softmax average in Year 1 ($t = 5.78, df = 6110, p = 7.94e\text{-}09$), and of `HWGen` versus `HWGen` group softmax average in Year 2 ($t = 4.74, df = 571, p = 2.61e\text{-}06$). However, `HWGen` and `HWGen` group vote were not found to be significantly different in Year 2, therefore we infer the order of performance to be `HWGen` group vote < `HWGen` < `HWGen` softmax averaging in Year 1, and `HWGen` softmax ave. < `HWGen` & `HWGen` group vote in Year 2.

5 Discussion

In its various modes, HWGen was able to more closely approximate the actions of teachers in the test dataset, when compared to the other policies tested here. This is despite the fact that both oracle policies have access to expert knowledge through the book ordering of material. In particular, the oracle-step policy makes use of both this ordering data and of user history to inform its selections. In contrast, for the HWGen policy, the neural network knows very little in advance. Instead it learns the ordering and pace of delivery from observations in training. Nonetheless the policy outperforms both oracles. This gives the approach considerable flexibility to be fitted to systems where a canonical ordering is not specified, since it can learn a suitable ordering itself.

In all, these results suggest that even when material is ordered linearly, an automated homework selection system like HWGen could be used to set homework in a more human-like way, both for individual students and for class groups. Thus our work may be viewed as a type of *recommender system* – concerned with selecting items for an individual based on their history in relation to others' histories [7, 15, 29] in a manner which will be familiar to users of many online services [19, 9, 13]. For online retailers the item bank in need of filtering is a set of products, for streaming services it is a library of movies and programmes, and for social networks it is user-generated content. In our case the item bank is a curated pool of physics and mathematics tasks aimed at school-children.

The recommender system approach has some precedent in educational technology. Early systems used heuristics, social networks and ontologies [3, 26, 32] before data-driven collaborative filtering techniques were introduced [31, 21, 35]. Here we show that deep behavioural cloning of teacher decisions is a viable method for homework selection in the educational domain. In future work we plan to implement HWGen as a live recommender system for the Isaac Physics tutoring platform.

6 Conclusion

In this work we introduced a method of automatic homework selection (HWGen), based on a deep neural network and trained on the behaviour of human teachers. We showed that HWGen was able to track the behaviour of previously unseen human teachers more closely (in RMSE terms) than baseline heuristic policies, including those with knowledge of the curriculum – despite HWGen having no access to such knowledge. We suggest this allows HWGen to be fitted to pre-existing systems, where historical data is available for training. Furthermore, with simple modification the HWGen approach can also set homework for groups of students, making it suitable for private study and classroom-based use.

It remains to be seen whether HWGen, either in individual or group mode, leads to improved learning for students. We will seek to address this question in future work with reward-based models.

References

1. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
2. R. Aboalela and J. Khan. Model of learning assessment to measure student learning: Inferring of concept state of cognitive skill level in concept space. In *Soft Computing & Machine Intelligence (ISCMI), 2016 3rd International Conference on*, pages 189–195. IEEE, 2016.
3. M. Anderson, M. Ball, H. Boley, S. Greene, N. Howse, D. Lemire, and S. McGrath. RACOFI: a rule-applying collaborative filtering system. In *Proceedings of IEEE/WIC COLA'03*, 2003.
4. H. Ba-Omar, I. Petrounias, and F. Anwar. A framework for using web usage mining to personalise e-learning. In *Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies*, 2007.
5. B. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13:4–16, 1984.
6. I. Bratko, T. Urbančič, and C. Sammut. Behavioural cloning: phenomena, results and problems. *IFAC Proceedings Volumes*, 28(21):143–149, 1995.
7. J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.
8. C. Brinton, R. Rill, S. Ha, M. Chiang, R. Smith, and W. Ju. Individualization for education at scale: MIIC design and preliminary evaluation. *IEEE Transactions on Learning Technologies*, 8:136–148, 2015.
9. R. Burke. Hybrid recommender systems: survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370, 2002.
10. M. Chi and K. VanLehn. Porting an intelligent tutoring system across domains. *Frontiers In Artificial Intelligence And Applications*, 158:551, 2007.
11. F. Chollet et al. Keras. `https://keras.io`, 2015.
12. S. D. Craig, C. Anderson, A. Bargagloitti, A. C. Graesser, T. Okwumabua, A. Sterbinsky, and X. Hu. Learning with ALEKS: The impact of students' attendance in a mathematics after-school program. In *International Conference on Artificial Intelligence in Education (AIED)*, pages 435–437. Springer, 2011.
13. M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22:143–177, 2004.
14. S. Dettmers, U. Trautwein, O. Lüdtke, M. Kunter, and J. Baumert. Homework works if homework quality is high: Using multilevel modeling to predict the development of achievement in mathematics. *Journal of Educational Psychology*, 102(2):467, 2010.
15. J. Dron, R. Mitchell, P. Siviter, and C. Boyne. CoFIND – an experiment in n-dimensional collaborative filtering. *Journal of Network and Computer Applications*, 23:131–142, 2000.
16. F. Khan, B. Mutlu, and X. Zhu. How do humans teach: On curriculum learning and teaching dimension. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1449–1457, 2011.

17. D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
18. R. V. Lindsey, J. D. Shroyer, H. Pashler, and M. C. Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological Science*, 25(3):639–647, 2014.
19. A. Munro, K. Höök, and D. Benyon. *Personal and social navigation of information space*. London: Springer-Verlag, 1999.
20. T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, pages 98–129, 1999.
21. R. Nadolski, B. van den Berg, A. Berlanga, H. Drachsler, H. Hummel, R. Koper, and P. Sloep. Simulating lightweight personalised recommender systems in learning networks: a case for pedagogy-oriented and rating based hybrid recommendation strategies. *Journal of Artificial Societies and Social Simulation*, 12, 2009.
22. J. D. Novak and A. J. Cañas. The theory underlying concept maps and how to construct and use them. *Technical Report, Institute for Human and Machine Cognition*, 2008.
23. Z. A. Pardos, S. Tang, D. Davis, and C. V. Le. Enabling real-time adaptivity in MOOCs with a personalized next-step recommendation framework. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 23–32. ACM, 2017.
24. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
25. D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.
26. M. Recker, A. Walker, and K. Lawless. What do you recommend? implementation and analyses of collaborative filtering of web resources for education. *Instructional Science*, 31:229–316, 2003.
27. Y. Rosen, I. Rushkin, R. Rubin, L. Munson, A. Ang, G. Weber, G. Lopez, and D. Tingley. The effects of adaptive learning in a massive open online course on learners' skill development. In *Proceedings of Learning @ Scale*, 2018.
28. D. Sampson and C. Karagiannidis. Personalised learning: Educational, technological and standardisation perspective. *Interactive Educational Multimedia*, 4:24–39, 2002.
29. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW'01*, 2001.
30. C. Sharp. Should schools set homework. *National Foundation for Educational Research*, 27:1–4, 2002.
31. T. Y. Tang and G. McCalla. Smart recommendation for an evolving e-learning system. *International Journal on E-learning*, 4:105–129, 2005.
32. J. Tarus, Z. Niu, and G. Mustafa. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50:21–48, 2018.
33. K. VanLehn. The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education (IJAIED)*, 16:227–265, 2006.
34. K. VanLehn, C. Lynch, K. Schulze, J. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The Andes physics tutoring system: five years of evaluations. *In Proceedings of the 12th International Conference on Artificial Intelligence in Education*, pages 678–685, 2005.

35. P.-Y. Wang and H.-C. Yang. Using collaborative filtering to support college students' use of online forum for english learning. *Computers & Education*, 59:628–637, 2012.