Massive Computerized Adaptive Testing

Mehdi Douch, Yassine Esmili, Sein Minn, Jill-Jênn Vie
Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9198-CRIStAL
F-59000 Lille, France
`{mehdi.douch,yassine.esmili,sein.minn,jill-jenn.vie}@inria.fr`

Abstract

We present a simple solution to Task 4 of the 2020 NeurIPS education challenge. This task falls within the personalized education category. Participants were asked to submit some code that could generate a sequence of 10 questions to ask a student in adaptive way, and then predict answers on a held-out set of questions. Participants were judged on the predictive accuracy of their model. Our solution, based on simple item response theory, and not using any metadata, got ranked 5th.

1 Rasch model

In the Rasch model (1961), each student is represented by a scalar θ_i that can be interpreted as their ability, and each question is represented by a scalar d_j which can be interpreted as its difficulty. R_{ij} represents a binary variable that indicates whether student i can solve correctly question j. The probability of success depends on the difference between the learner ability and question difficulty:

$$\Pr(R_{ij} = 1) = \sigma(\mu + \theta_i - d_j)$$

where μ is a bias parameter and $\sigma : x \mapsto 1/(1 + \exp(-x))$ is the sigmoid function. This model is usually trained by minimizing the cross-entropy loss, and can be implemented efficiently as logistic regression with sparse features, as recommended by Vie and Kashima (2019).

2 Proposed approach

To make a computerized adaptive test, we need to choose the parameters initialization of unseen students, the strategy for choosing the next question, and the update rule of our model. We now describe those strategies for the online and batch models.

Initialization On the training data of existing 3934 students answering 948 questions, we calibrate the Rasch model to estimate student ability parameters and question parameters, with regularization parameter λ. For each of the 984 unseen students, we set their ability estimate to the mean of known ability estimates. For our submitted batch model though, we used the median.

Select the next question To choose the next question, we pick the question of probability closest to 0.5. This threshold can be fine-tuned.

Update To update our model, two approaches were tested. In the batch model, we retrain the Rasch model from scratch with all samples at hand. In the online model, question parameters are frozen and we just need to update the θ estimate for each new student given their answers so far. In this online case, the log-likelihood has a simple form, and can be optimized using Brent's method (1973) available in `scipy.optimize` (Virtanen et al., 2020).

1st NeurIPS Education Challenge (NeurIPS 2020), Vancouver, Canada.

Table 1: Accuracy results of our adaptive test algorithms.

Model	Elapsed time	Local valid	Public test	Private test
Online model	38 s	0.66968	0.6838	0.6846
Fine-tuned online model	25 s	0.67212		
Batch model	6 min 41 s	0.68901	0.6994	0.7103
Fine-tuned batch model	1 min	0.69357		

Fine-tuning The regularization parameter of the logistic regression, the fact of fitting or not a bias in the model, the threshold for the logistic regression are all hyper-parameters.

Our implementation of the Rasch model is based on scikit-learn (Pedregosa et al., 2011). All code is on GitHub[1].

3 Results

The experiments were done on a Core i7 1.90 GHz. Results are reported in Table 1. Our online model got a public accuracy of 0.6846, ranking 9th, while our batch model got a public accuracy of 0.7103, ranking 5th.

We could fine-tune our online model by imposing a L_2 regularization $\lambda = 1$ on the student and question parameters. For both online models, the bias μ was fixed to 0 and we used 0.54 as threshold for the logistic regression, as the metric chosen by the competition organizers was accuracy and not a metric balancing false negatives and false positives.

The batch model used a regularization parameter $\lambda = 1$ for each step of logistic regression. Curiously, the best fine-tuned batch model was obtained when the initial question parameters were not regularized $\lambda = 0$, but when there is regularization $\lambda = 1$ for all subsequent updates. Also, a positive bias μ was learned but not used for prediction, leading to underestimates in the probabilities of success, but this was consistently better (it is equivalent to tuning the item selection criterion and threshold of the logistic regression).

4 Discussion and Future Work

Batch models perform better and it is not surprising: at each step, the answers from all 984 students are observed and we can take advantage of the fact that this information is available to select the next question for all of them. This data is precious to improve the performance.

Even simple, the Rasch model is a strong baseline. We wish we could estimate multidimensional item response theory models, for example using factorization machines (Vie and Kashima, 2019). Their estimation sure would fit within the thirty allotted minutes for Task 4. Variational item response theory (Wu et al., 2020) is also a promising approach for computing approximate posteriors, and testing new criteria for item selection.

Acknowledgments and Disclosure of Funding

Mehdi Douch and Yassine Esmili, while working at Inria, are at the same time developing an entrepreneurial project focusing on adaptive learning for mathematics (Axiome).

References

R. P. Brent. *Algorithms for Minimization without Derivatives.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

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 É. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, Oct. 2011. URL `https://hal.inria.fr/hal-00650905`.

[1] `https://github.com/jilljenn/ktm/tree/jj/neurips-questions/starter_kit/task_4/`

G. Rasch. On General Laws and the Meaning of Measurement in Psychology. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 4: Contributions to Biology and Problems of Medicine*, pages 321–333, Berkeley, Calif., 1961. University of California Press. URL `https://projecteuclid.org/euclid.bsmsp/1200512895`.

J.-J. Vie and H. Kashima. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence*, pages 750–757, 2019. URL `https://arxiv.org/abs/1811.03388`.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. SciPy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020. URL `https://doi.org/10.1038/s41592-019-0686-2`.

M. Wu, R. L. Davis, B. W. Domingue, C. Piech, and N. Goodman. Variational item response theory: Fast, accurate, and expressive. In *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, pages 257–268, 2020. URL `https://educationaldatamining.org/files/conferences/EDM2020/papers/paper_22.pdf`.