Desperately Seeking Subscripts: Towards Automated Model Parameterization

J. MOSTOW
Y. XU
M. MUNNA
Carnegie Mellon University, USA

This paper addresses the laborious task of specifying parameters within a given model of student learning. For example, should the model treat the probability of forgetting a skill as a theory-determined constant? As a single empirical parameter to fit to data? As a separate parameter for each student, or for each skill? We propose a generic framework to represent and mechanize this decision process as a heuristic search through a space of alternative parameterizations. Even partial automation of this search could ease researchers' burden of developing models by hand. To test the framework's generality, we apply it to two modeling formalisms – a dynamic Bayes net and learning decomposition – and compare how well they model the growth of children's oral reading fluency.

Key Words and Phrases: model parameters, heuristic search, knowledge tracing, learning decomposition, children's reading fluency growth

1. INTRODUCTION

This paper addresses the problem of defining parameters, more precisely how specific to make them. For example, the parameters in a knowledge tracing model are the probabilities of already knowing a skill, learning it from a practice opportunity, guessing an answer without knowing the skill, or answering incorrectly despite knowing the skill. But how specifically should these parameters be defined? Should we use a different parameter for every skill? For every student? For every <student, skill> pair? The last option would generate too many parameters to fit from the available data. Corbett et al. [Corbett and Anderson, 1995] decided to make the knowledge parameters (probabilities of knowing already or learning) skill-specific, and the performance parameters (probabilities of guessing or slipping) student-specific. They judged that the knowledge probabilities vary more by skill than by student, whereas the performance probabilities vary more by student than by skill.

Such decisions – how specific to make a given parameter in order to predict unseen data – are the focus of this paper. This subtle but crucial modeling decision is typically made by hand, often by trial and error. The researcher explores various alternatives, trading off theoretical plausibility, computational tractability, model fit, statistical reliability, interpretability, and informativeness with respect to the research questions of interest. This problem falls in the domain of model selection but differs from prior work on selecting structure [e.g., Madigan and Raftery, 1994] or variables [e.g., Negrin et al., 2010] in that we focus on selecting a specific parameterization of the given variables.

We propose a generic framework to represent and mechanize this process. To test its generality, we apply it to two types of student learning models (dynamic Bayes nets and learning decomposition), which we train and test on children's oral reading fluency data.

Authors' addresses: J. Mostow, E-mail: mostow@cs.cmu.edu; Y. Xu, E-mail: yanbox@cs.cmu.edu; M. Munna, E-mail: mmunna@cs.cmu.edu. Project LISTEN, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, U.S. This work was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080628 to Carnegie Mellon University. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or U.S. Department of Education.

2. A HEURISTIC SEARCH SPACE OF MODEL PARAMETERIZATIONS

The title of this paper refers to model development as a search for subscripts because subscripts indicate the specificity of the parameters they index. To formalize this search space, we represent each state in the space as a vector with an element for each parameter in the model. For example, consider a dynamic Bayes net model of Knowledge Tracing (KT), with probabilities for *guess*, *slip*, *forget*, *learn*, and *already know*. We represent a parameterization of this model as a vector of 5 elements, each of which specifies how the corresponding parameter is subscripted. For readability, we write the value of each element as a phrase describing how the parameter is indexed, e.g. *'by student'*, *'by skill'*, *'by student level'*.

Formally, we define a **parameterization** of a model with m parameters $p_1, p_2, \ldots, p_m$ as a vector of m split functions ($F_1, F_2, \ldots, F_m$), each of which specifies how to index the corresponding parameter over a set of size N, which we call the *size* of the split. For example, to fit the *guess*, *slip*, and *learn* parameters of a KT model separately for each student, we use the '*by student*' function to split them into separate parameters $guess_j$, $slip_j$, and $learn_j$ for each student j, so its size is the number of students. Likewise, to fit the *already know* parameter separately to the data for each skill, we use the *'by skill'* function to split it into separate $already\ know_i$ parameters for each skill i, so its size is the number of skills.

To set a parameter to a single value for all of the data, we use a function named *"by NULL"* to leave the parameter as is, with no subscripts or splits. We may estimate its empirical value by fitting the data, or supply a theoretical constant. For example, for a KT model, we apply the *"by NULL"* function to the *forget* parameter, and set its value to zero based on the theoretical assumption of no forgetting.

We define the size of a parameterization as the summed sizes of its split functions. Intuitively, this quantity is simply the total number of subscripted parameters. The example parameterization above indexes three parameters by student and one by skill, so its size is 3 * # students + 1 * # skills.

Given m parameters $p_1, p_2, \ldots, p_m$ and a set F of split functions, the cross product F^m generates a search space of $|F|^m$ possible model parameterizations to consider. One simple but inefficient search strategy is brute force, searching for the best model over all expressible splits. Alternatively, one heuristic strategy is to search the space of parameterizations in order of increasing size, fitting the resulting parameterized model to the data, computing some measure of its (complexity penalized) model fit, and halting when we reach a local maximum. Note that the size of the parameterization is a crude measure of model complexity.

3. TWO DIFFERENT MODEL FORMALISMS

Dynamic Bayes nets (DBNs) provide a powerful way to infer a student's changing knowledge over time from observed student behavior. We extended a previous DBN model of children's fluency growth [Beck et al., 2008] by adding an observable "Distributed Practice" node whose value is 1 for the student's first encounter of the day for a given word and 0 otherwise. The resulting model (shown in Fig. 1) has 17 parameters, too many to list here. For example, the parameter "*learn* | *distributed practice, help*" models the probability P(K_n = *true* | K_{n-1} = *false*, Dn_{-1} = *true*, H_{n-1} = *true*). We used BNT-SM [Chang et al., 2006] to express different parameterizations of the model and fit them to data.

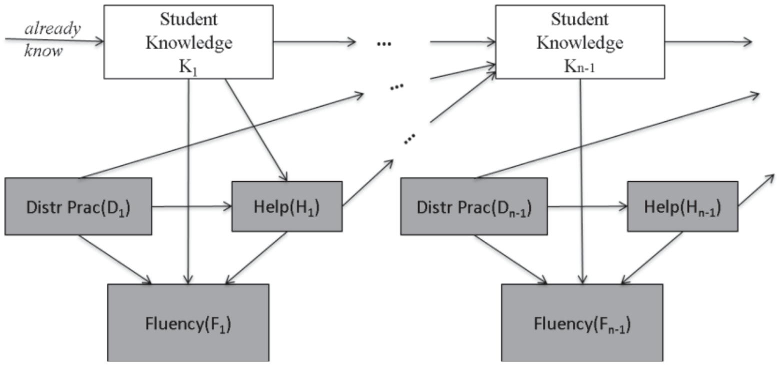

Fig. 1. Architecture of a Bayes Net Model of Children's Growth in Oral Reading Fluency

Learning decomposition (LD) estimates the relative impact on performance of different types of practice, such as wide vs. repeated reading and distributed vs. massed practice [Beck, 2006]. Using this approach, we developed the following model to predict a child's latency prior to reading a word aloud in text:

$$latency = E + L * word\_length + A * e^{-b*(h*m*HM + h*HD + m*NHM + NHD)}$$

Here ***E*** represents minimum latency, ***L*** scales latency as a linear function of word length, ***A*** reflects the latency at the first encounter of a word, and ***b*** represents the learning rate. The coefficient ***h*** represents the impact of a tutor-assisted encounter relative to an unassisted encounter. The coefficient ***m*** represents the impact of a massed encounter (i.e. of a word seen earlier that day) relative to a distributed encounter (i.e. of a word seen for the first time that day). The variable ***HM*** counts the number of assisted, massed encounters; ***HD*** counts assisted, distributed encounters; ***NHM*** counts unassisted, massed encounters; and ***NHD*** counts unassisted, distributed encounters. To fit different parameterizations of this model to data, we used MATLAB's (Ver. 7.6.0.324) non-linear regression function.

4. EVALUATION

4.1 Data

The oral reading fluency data for this paper comes from a random sample of 40 children, stratified by gender and reading level, from the students who used Project LISTEN's Reading Tutor [Mostow and Aist, 2001] during the 2005-2006 school year, with a median usage of 5.7 hours. In total they attempted to read 5,078 distinct word types ranging in difficulty level from grades 1 to 11. The data includes each student's unique user id, gender, reading level (from grade K to 6), and performance on each word encounter, which we define as fluent if accepted by the Reading Tutor as read correctly without help or hesitation.

To partition the data into training and test sets, we ordered the distinct word types encountered by each student by the number of encounters. We assigned all the student's encounters of odd-numbered word types to the training set, and all encounters of even-numbered word types to the test set, so as to be able to train and test models on all of a student's encounters of a given word.

Given the information in the data set, one set of possible splits is *{'by student', 'by student level', 'by gender', 'by word', 'by word level', 'by student and word level', 'by* *student level and word', 'by student level and word level', 'by gender and word', 'by gender and word level'*}. We omitted the split *'by student and word'* because we had no overlap in <student, word> pairs between training and test sets.

4.2 Results

Table I compares different parameterizations of DBN and LD models, ordered by size. The DBN models treat fluency as a binary variable, so we show the percentage accuracy of their predictions, both overall and within-class; the test data is unbalanced, with 72% of it in the positive (fluent) class. The LD models predict real-valued latencies, so we use Root Mean Squared Error (RMSE) to measure their accuracy. Since the models make different types of predictions, their accuracies are not comparable. Given the maximized value L of the likelihood function for the estimated model, the number k of parameters and the number n of data points in the training set, we compute AIC (Akaike Information Criterion) as $AIC = 2k - 2\ln L$. We estimate BIC (Bayesian Information Criterion) as $BIC \approx -2 \cdot \ln L + k \ln(n)$.

DBN and LD models use different likelihood functions. The likelihood function for DBN models is a probability, so their AIC and BIC scores are positive. In contrast, the likelihood function for a linear regression is a product of Gaussian probability density functions, so AIC and BIC scores for LD models can be positive or negative.

Table 1. Accuracy and complexity of DBN and LD models on unseen test data for children's oral reading fluency. The best value(s) in each column are underlined.

Model: split by…	DBN						LD			
	Size	Acc (%)	Acc on +	Acc on -	AIC	BIC	Size	RMSE (sec)	AIC	BIC
NULL	17	72.5	99.7	2.4	504084	504243	6	0.22	-9056	-9000
gender	34	72.5	99.7	2.4	504060	504377	12	0.22	-9269	-9157
student level	136	72.5	99.7	2.4	897462	898733	42	0.23	46	427
word level	170	72.5	99.7	2.5	474230	475818	48	0.35	-10201	-9755
gender, word level	323	72.5	99.7	2.5	474182	477200	96	0.32	-36957	-36048
student	680	72.6	98.1	7.1	497074	503427	210	0.21	-22483	-20546
student level, word level	1054	72.5	98.1	6.5	470057	479905	318	0.28	25977	28692
student, word level	4573	71.8	94.1	14.2	473541	516270	1512	0.18	-144840	-130160
word	5848	72.4	96.0	1.2	495727	550370	1518	0.20	-9136	4715
gender, word	11271	72.5	93.7	15.2	550977	847543	2856	0.18	-32541	-5762
student level, word	31739	71.8	98.1	6.5	512257	617572	3588	0.21	15354	45053

Which models are best? None of the DBN models substantially beats the majority class accuracy of 72%. The five simplest models have almost perfect recall (accuracy on positive examples), but very low accuracy on negative examples. Note that AIC and BIC do not vary smoothly with the size of the parameterization. For example, splitting by student level has size 136 and gives the worst AIC and BIC scores, while word level has size 170 but yields the best BIC score and a near-best AIC score.

The *'by student and word level'* LD model has the lowest AIC and BIC scores. This fact suggests that students at the same estimated student level differ enough to model individually, possibly due to inaccurate estimates. In contrast, word level apparently captures adequate information about word difficulty.

This model also achieves the best accuracy on unseen test data (RMSE = 0.18 sec). However, the second best accuracy is achieved by *'by word'* model, which has some of the worst AIC and BIC scores even though its size is not enormously larger (1518 vs. 1512). This disparity implies that AIC and BIC can be poor predictors of performance on unseen data. One problem we faced that due to splitting when the dataset size was very small (e.g. less than 4 data points) we failed to fit the LD model. We excluded these datasets and the size of parameterization became smaller than it should be in some of the models.

Although the DBN and LD models have different formalisms and outputs, they are not directly comparable, we can still compare their performance profiles over the same space of parameterizations. In particular, is the same parameterization best for both models? No. For the LD models, the *'by student, word level'* parameterization achieves by far the best AIC and BIC scores. For the DBN models, this parameterization achieves close to the best AIC score, which is for the *'by student level and word level'* model, but so do the *'by word level'* and *'by gender and word level'* models. Moreover, its BIC score is mediocre.

5. CONCLUSION

This paper defines the problem of parameterization selection and formalizes it in terms of a space of parameterizations induced by split functions. It proposes a simple strategy to search this space in order of size, hill-climbing on complexity-penalized model fit. We implemented a prototype of this strategy restricted by using the same split function for every parameter to accommodate a limitation of BNT-SM. We demonstrated its generality by applying it to both DBN and LD models and evaluating the resulting parameterizations on the same data set.

Future work includes expanding the search space to relax the restriction in the implementation, and devising search heuristics to go beyond size and complexity-penalized model fit and address additional criteria discussed in the Introduction. This work will succeed if it helps clarify, accelerate, or automate the discovery of good models.

REFERENCES

BECK, J.E. 2006. Using learning decomposition to analyze student fluency development. In *ITS2006 Educational Data Mining Workshop*, Jhongli, Taiwan, June 26, 2006, C. HEINER, R. BAKER and K. YACEF, Eds., 21-28.

BECK, J.E., CHANG, K.-M., MOSTOW, J. and CORBETT, A. 2008. Does help help? Introducing the Bayesian Evaluation and Assessment methodology. In *9th International Conference on Intelligent Tutoring Systems*, Montreal, June 23-27, 2008, 383-394. ITS2008 Best Paper Award.

CHANG, K.-M., BECK, J., MOSTOW, J. and CORBETT, A. 2006. A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, Jhongli, Taiwan, June 26-30, 2006, K. ASHLEY and M. IKEDA, Eds., 104-113.

CORBETT, A. and ANDERSON, J. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 253-278.

MADIGAN, D.M. and RAFTERY, A.E. 1994. Model selection and accounting for model uncertainty in graphical models using Occam's Window. *Journal of the American Statistical Association* 89, 1335-1346.

MOSTOW, J. and AIST, G. 2001. Evaluating tutors that listen: An overview of Project LISTEN. In *Smart Machines in Education*, K. FORBUS and P. FELTOVICH, Eds. MIT/AAAI Press, Menlo Park, CA, 169-234.

NEGRIN, M.A., VAZQUEZ-POLO, F.J., MARTEL, M., MORENO, E. and GIRON, F.J. 2010. Bayesian Variable Selection in Cost-Effectiveness Analysis. *International Journal of Environmental Research and Public Health* 7, 1577-1596.