Statistical Techniques For Comparing ACT-R Models of Cognitive Performance

Ryan Shaun Baker (rsbaker@cmu.edu)
Albert T. Corbett (corbett@cmu.edu)
Kenneth R. Koedinger (koedinger@cmu.edu)
Human-Computer Interaction Institute, Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA 15213 USA

Abstract

We discuss how to apply statistical tests to compare different ACT-R models, by treating the ACT-R models as approximations of the mathematical models underlying them. To this end, we propose a method for deciding how many times to run a model, and a method for determining how many free parameters each model has.

Introduction

Usually, there is more than one possible account or model for a phenomena, or for a set of phenomena. Sometimes a research group creates multiple models varying along some dimension of theoretical interest – at other times, one group of researchers wants to compare their model to a previously published model. In either case, we must determine which model better fits the data.

Two broad approaches have been taken to compare the quality of different models. The first is to compare the two on new data. This can be a second data set taken for the same task and general population, or data from a moderately distant transfer task testing each model's generality. Choosing a transfer task can be tricky. If a specific transfer task is chosen, and model A is found to be better than model B, there is no guarantee that the opposite would not be true on another transfer task. Nonetheless, it can be an effective method for comparing how well the models generalize, and has been used in "competitions" between different cognitive architectures. (Gluck and Pew 2002)

Alternatively, models can be compared within the original data set. In comparing two models, both the absolute fit to the data and the flexibility of the fitting techniques must be taken into account. Computational models have been criticized for not taking full account of the relative complexity of the models being compared (Roberts and Pashler 2000) – and to address this concern, cognitive modelers have been paying increasing attention to model selection formulas developed by the statistical community to explicitly deal with the number of free parameters, (Zucchini 2000) and the cross-validation techniques used in the machine learning community. (Mitchell 1997).

Statistical Techniques For Model Comparison

In their extensive review of methods for evaluating goodness of fit, Schunn and Wallach (in press) conclude that model selection formulas, despite their many advantages, are too difficult for most computational modelers to use to compare current computational models. They point in particular to the difficulty of developing closed-form equations for a computational model, and the difficulty of determining the proper number of free parameters and their relative impact, in the absence of closed-form equations.

One answer to the need for closed-form equations is to develop them. Developing such equations is possible for ACT-R models of performance that always terminate in a specific set of behaviors, as ACT-R's behavior is based on a specific set of well-defined closed-form equations. The specific equations explaining the proportions of behaviors in a given model can therefore be developed using these equations and the productions' inputs and outputs.

This approach has been conducted in the past (Anderson and Matessa 1997), but Schunn and Wallach are correct that it becomes extremely difficult and time-consuming for even moderately complex models. For example, 55 equations are needed to represent a moderately complex model of early algebra problem-solving, with as many as 44 terms in a single equation. (Koedinger and MacLaren 2002)

In this paper, we discuss how it may not be necessary to actually derive equations in order to conduct statistical analysis on ACT-R models, and present a case-study of using ANOVA and BiC (the Bayesian Information Criterion) to compare computational models.

In order to present a tractable first discussion of this topic, we will limit the scope of this paper to models of cognitive (as opposed to perceptual and motor) performance at a specific stage in the learning process, referred to as "terminal models" by Salvucci and Anderson (1997). Such models reflect the assumption that the behavior being studied is sufficiently developed that it is not changing during the course of investigation – an assumption Anderson, Lebiere, and Lovett (1998) refer to as a "typical assumption in much of the experimental research on human cognition", justified "in cases where the behavior under study is at some relatively asymptotic level or the crticial factors being investigated do not change over the range of experiences encountered in the experiment." In other words, these models may involve the creation of new memory chunks, but do not involve the creation of new productions or changes in productions' utilities. A sizable minority of ACT-R 4.0 and 5.0 models fit this assumption, including models of performance at the Tower of Hanoi task (Anderson and Lebiere 1998), models of the fan effect (Anderson and Reder 1999), and models of student performance at educational tasks (Koedinger and MacLaren 2002), (Nokes, Ohlsson, and Corrigan-Halpern 2002).

We believe that the methods presented here can be extended in fairly straightforward ways to models relying upon ACT-R's perceptual and motor modules, models where utility parameters shift over time, and even models where new productions are created – but we leave this for future work.

Computational Models Approximate Mathematical Models

In effect, when a computational model is run once, it gives an approximation of the mathematical equations that can be used to describe it. (Simon 1992) By running it a greater number of times, it produces a more accurate approximation of the solution of those equations. As the number of runs approaches infinity, the error of the approximation will reach 0.

The question then becomes how many times the model should be run to appropriately approximate the mathematical estimates of the data. One conservative strategy is to make sure the model's results will fall within a certain confidence interval a pre-selected percentage of the time, given the worst-possible standard deviation s of results (which will be the square root of half the the range of the possible values of the data – which is $\sqrt{0.5}$ for data consisting of proportions of results expressed between 0 and 1).

The equation for computing the desired sample size (n) is expressed in terms of the desired percentage of the time the model will be within the given range (α), the value of the t-distribution corresponding to α, given the sample size $(t_n(\alpha/2))$, the distance allowed in either direction from the actual proportion (d), and s. Using the standard equation for confidence intervals for a t-distribution[1], we find:

$$d = (t_n(\alpha/2))\frac{s}{\sqrt{n}} \quad \text{or} \quad n = \left((t_n(\alpha/2))\frac{s}{d} \right)^2$$

In choosing how tight to make the confidence intervals, it is worth considering how tight the confidence intervals of the data set are. There is no penalty for making the model's estimates arbitrarily tight (except for time), but there is also not much need to make those estimates orders of magnitude tighter than the estimates in the original data set. If that level of precision is needed to compare different candidate models, there is considerable risk of determining which model better fits the error in the data rather than which model better fits the data itself.

So, for example, if it was judged appropriate to make sure that every proportion will be found within 5% in either direction 95% of the time (p=0.05 that it will be outside that range), then using the values (d = 0.05, α=0.05, s=$\sqrt{0.5}$), n should equal at least 778. Hence, we recommend modelers desiring this level of precision run their model 778 times when making final calculations of the model's fit to the data.

In principle, a lower minimum n might be found by taking into account the level of stochasticity in the model to determine a lower value for s, but in practice this is not necessary, since the method presented here produces conservative but tractable minimum bounds on the number of runs necessary.

After running the model an appropriate number of times to closely approximate the closed-form equations, we can treat the model's results the same way we would treat what would result from closed-form equations. By taking the difference between its predictions and the data values, we can compute residuals.[2] These can then be used to make model comparisons. The other piece of information which will be needed to conduct these analyses is the number of free parameters each model uses, which will be discussed in the following section.

Assessing Model Complexity: How Many Free Parameters Are Needed?

When comparing two models of data, it is important not just to compare the closeness of their fit to the data set but their comparative complexity. The more complex a model is, the more likely it can closely fit an arbitrary data set, or the error in that data set, by chance. This limits that model's generalizability, a phenomenon usually termed "overfitting". To address this, several methods have been developed for assessing the comparative complexity of different models, and the interaction between this and their goodness-of-fit.

Some approaches to computing complexity, such as Minimum Description Length (MDL), take into account the relative influence of different factors on the number of potential fits the model can make (Pitt et al 2002). Other methods, such as the Bayesian Information Criterion (BiC), use a more approximate measure of complexity, by identically treating each factor (termed a parameter) that can affect the model's results, and counting the number of these parameters for each candidate model (Raftery 1995). The debate between these two strategies for complexity analysis is currently very active in the statistical community. In this paper, we will be following the parameter-counting approach, as it offers substantial information and is much easier to conduct in the absence of closed-form equations.

There are three potential sources of free parameters in an ACT-R model: its productions, its chunks, and its ACT-R global parameters. In the following sections, we will discuss how to count the parameters from each of these sources.

Productions

In order to determine how many free parameters can be accounted for from the productions, we need to analyze the equations that underlie ACT-R 5.0. (ACT-R Research Group 2002) In ACT-R, the likelihood that any production will be used is based on the production's utility, $U_i = \rho_i G - C_i + \varepsilon$, where ρ_i stands for the (expected) probability that firing the production will lead to correctly completing the current objective, G stands for the value of the objective, and C_i stands for the expected cost of accomplishing the objective.

[1] By the central limit theorem, the estimates of the mean and variance should be approximately normally distributed for large samples, even if the population (of the results of the ACT-R model) has a very different distribution, (Stilson 1966) allowing us to use the t distribution. In cases where ACT-R's behavior is extremely skewed and long-tailed, a phenomenon observed in ACT-R models involving utility learning (Young and Cox 2002), transformation methods can be used to increase the sample's normality. (Ramsey and Schafer 1997) This should *not* be a substantial problem, however, for the performance models discussed here.

[2] It would be valuable to relate the uncertainty in these residuals to the uncertainty captured by the various goodness-of-fit/flexibility-of-fit criteria we discuss later in the paper, and this is an area we intend to investigate. At this point, though, it is sufficient to note that the uncertainty of these residuals can be made substantially smaller than the difference in uncertainty between the models.

ε stands for the noise added to the result (in order to determine what the proportions of different results will be), and is calculated using a logistic distribution with a mean of 0 and a variance determined using a global parameter, s. Given expected utility U_i, the probability that a given production will fire at any given point is computed as follows, where j ranges over all productions that could fire at this point:

$$P(Pi) = \frac{e^{\frac{U_i}{\sqrt{2}s}}}{\sum_j e^{\frac{U_j}{\sqrt{2}s}}}$$

For example, when there are two productions that could fire, the following equations are used:

$$P(P1) = \frac{e^{\frac{U_1}{\sqrt{2}s}}}{e^{\frac{U_1}{\sqrt{2}s}} + e^{\frac{U_2}{\sqrt{2}s}}}, P(P2) = \frac{e^{\frac{U_1}{\sqrt{2}s}}}{e^{\frac{U_1}{\sqrt{2}s}} + e^{\frac{U_2}{\sqrt{2}s}}}$$

In general, when computing the probability a specific behavior will be expressed, it is necessary to multiply together the probabilities of each production in each chain of productions that produces that behavior. Thus, if result A is produced solely by production P1, P(A) = P(P1). If result A is produced by productions P1 and P3 in combination, or by productions P2 and P4 in combination, then
P(A)= P(P1)*P(P3|P1) + P(P2)*P(P4|P2).

Therefore, the probability of behavior A depends at least on every production that could have fired to produce behavior A. It also depends on all of the other productions that could have fired at those steps and produced a different result, as those productions' utilities are used in the denominator of the probability of each production. Hence, each of the productions that could have fired at those steps must be counted as a free parameter.

Given this, our strategy for estimating the number of parameters more or less follows Simon's (1992) suggestion that every production be counted as a free parameter. However, we recommend a few refinements on this general approach. For instance, some productions do not need to be counted as free parameters. Such productions fall into two categories: First, productions which do not affect the results which will be compared to data. Almost every model will have a few productions that are essential to the implementation but are not part of the model of knowledge: productions that handle book-keeping, productions that prepare the model for another run, and so on. Generally, these productions occur every run or cycle, or always co-occur with other productions – leading us to our second category. If two productions P1 and P2 always co-occur (after P1 fires, P2 always fires – it never fails to fire, and there is no other production that could fire in its stead), they can and should be counted as only one free parameter -- even if there is a set of productions that fire in between P1 and P2. Co-occurrence can be determined during model design, by inspection, or via post-hoc sensitivity analysis. (Koedinger and MacLaren 2002)

Beyond these cases, every production should be counted as at least one free parameter. Even if two productions are yoked together to have exactly the same ρ, each production's existence can produce qualitatively different behavior and affects the utility of the other productions.

There are even cases where a production should be counted as two parameters. If both of a production's terms involved in computing utility -- ρ_i and C_i -- are allowed to float, then that production should be counted as two free parameters. If only ρ_i or C_i, or neither of the two, is defined for the production, then it will count as one free parameter.

Memory

Similar analysis can be applied to calculating the number of free parameters accounting for the declarative chunks within a given model. During declarative retrieval, the activation of any given chunk i equals its Base-Level Activation (B_i) plus the total spreading activation given by other chunks. The spreading activation of a given chunk j, written W_jS_{ji} in the equation below, is the product of W_j, which equals the global activation parameter Ga, divided by the number of chunks that j references.

$$A_i = B_i + \sum_j W_j S_{ij}$$

This formula is then used to compute the probability of retrieval and the latency taken to retrieve the chunk.

$$P(i) = \frac{e^{\frac{A_i}{\sqrt{2}s}}}{\sum_j e^{\frac{A_j}{\sqrt{2}s}}} \quad , \quad RT(i) = Fe^{-A_i}$$

As can be seen, the formula for the probability of retrieving a chunk is the same as the formula of the probability of choosing a matching production, except for the substitution of activation for utility. Calculations of latency require the same information as calculations of probability of retrieval, as such calculations rely upon activation and the production having already been retrieved.

Determining the number of free parameters given by the declarative chunks thus relies on the free parameters used in determining activation, which includes all of the chunks that could have been recalled. Since activation includes spreading activation, it also includes every chunk that spreads activation to one of those chunks. Hence, every chunk that can be retrieved, or that spreads activation to a chunk that can be retrieved, should be counted as a free parameter.

Note that this does not just apply to chunks that existed at the beginning of the model's run. If the model creates declarative chunks during its run, these chunks need to also be included in the counts of free parameters. In general, every unique chunk that is created on any run should be counted as a free parameter – each chunk's existence or non-existence on any specific run certainly affects the equations that describe the model's performance. Two chunks can be considered unique if there are any situations where one would be retrieved or spread activation, and the other would not.

As with productions, some chunks do not need to be counted. If there is a chunk which is only used for information storage rather than to produce the pattern of results in the model, and it spreads no activation, it can be excluded. Such a chunk must necessarily fulfill two conditions: its retrieval never fails, and there is never a case where it is competing with another chunk for retrieval. Additionally, a chunk can be removed from consideration as a parameter if its slots do not change and it is associated one-to-one with a specific preceding production – the production always leads to the chunk being retrieved, not to a failure or another chunk.

ACT-R Global Parameters

The third source of free parameters is ACT-R's global parameters. We propose here that ACT-R global parameters be treated as free parameters and given the same weight as productions and chunks (and will discuss the limitations to this approach in the next section). However, not all ACT-R global parameters that exist need to be counted as free parameters. Any parameter chosen before any model-fitting is attempted can be treated a constant rather than as a free parameter.

The fact that some global parameters can be excluded from consideration necessarily calls for honesty on the part of modelers as to what parameters were allowed to vary at any point, and which were chosen beforehand; but this should be easy to discern. In practice, if a parameter is left at ACT-R default values, at 0, or at a well-known parameter derived from previous experiments (as in Lebiere and West, 1999 and Lebiere, Wallach, and West, 2000) and its value was never manipulated, than it can be omitted from the list of free parameters. But if it was ever tweaked, it should be treated as a free parameter.

Summary: Computing the number of free parameters

For terminal ACT-R models, we recommend the following approach (given the caveats discussed in the section above):

- Use a minimum of one parameter per production used during the steps of interest. If both Pi and Ci vary, use two.
- Use one parameter per memory element which is used in the steps of interest, and which either competes with another chunk or can fail to be retrieved. Include a parameter for any other memory element that spreads activation to one of the memory elements that can be retrieved in the steps of interest.
- Use one parameter per ACT-R global parameter allowed to vary.

Again, we believe it is both possible and desirable to extend this approach both to models which learn, and models with radically different ratios of different types of parameters. We leave this to future work.

We conclude by again reminding our readers to carefully document what productions and memory chunks are treated as free parameters. When comparing two models, especially those produced by different researchers, it is of paramount importance that free parameters are counted in the same fashion for each model.

A Case Study in Model Comparison

We have had the opportunity to explore some of these ideas in comparing computational models of student errors in constructing scatterplots of data (Baker, Corbett, and Koedinger 2001,2002a), in order to inform the design of a cognitive tutor (Baker et al 2003)

Scatterplots should contain the relationship between two quantitative variables, but when students were given two such variables, plus a categorical variable as a distractor, students frequently committed two conceptually similar errors. When given no advice on which variables to place in their graph, 15% made what we call the *variable choice* error, incorrectly choosing a categorical variable for the X -- 0% used the correct variables. Naming the variables to use in the question did not eliminate this error, but 77% used the correct variables. 13% of those students, however, then made what we term the *nominalization* error: treating the values of the quantitative X variable as if they were categorical. They wrote the variable's values along the axis in the order they appeared in the data table, rather than numerical order, e.g., placing "22 20 23 25 24 19 23" along the axis rather than "19 20 21 22 23 24 25". It was also found that labeling the axis variables for the student did not significantly reduce the representation error's frequency. The frequency of these errors is shown in Table 1.

Given the conceptual similarity between these two errors, we wondered if they could be explained as execution of the same strategy or as the execution of different strategies producing similar results. We were also interested in determining what type of behavior underlied correct performance in this domain, and what the role of factors such as the variables in the question was.

Fitting and comparing models of the data

We created a set of ACT-R models that represented this data, and compared their ability to fit the data. For each model, we used multiple runs with different starting points of an iterative gradient descent algorithm (courteously provided to us by Christian Lebiere) to find the best possible parameters. During runs of IGD, we minimized a function combining r^2 and the Mean Absolute Deviation (MAD). To compute each model's predictions at each step of the process, we ran every condition of each model 778 times.

The data and the predictions of our models were represented as the proportions of occurrence of each behavior, with the probabilities of the events of the second step as probabilities contingent on correct behavior on step 1. (This revealed that there were no observations for step 2 in the no prompts condition, because no student made it to step 2. Thus, we excluded those cells during data fitting.)

Only 3 global ACT-R parameters were allowed to vary: the utility (:ut) and retrieval (:rt) thresholds, and the expected gain (:egs). We had 6 declarative chunks that could be sometimes retrieved in place of one another, giving six more parameters. Since base-level activation was set much higher than the retrieval threshold, failure to retrieve a memory chunk did not occur, and the other chunks did not need to be counted as parameters. We allowed the ρ of the productions which produced strategic decisions to vary, but in accordance with the policy decided on earlier, counted every production used in the steps of interest as a parameter, except for productions which always fired – and only fired – after another specific production had fired.

	No prompts	No labels	X variable labeled	Y variable labeled	Both variables labeled
Variable choice error	15.0	26.9	7.7	26.9	6.5
Correct axis variables (CAV)	0	73.1	79.3	73.1	77.4
Given CAV, nominalization error on X axis only	n/a	15.7	17.4	15.7	12.5
Given CAV, nominalization error on Y axis only	n/a	0	0	0	0
Given CAV, nominalization error on both axes	n/a	5.3	8.7	0	8.3
Given CAV, correct variable representation on each axis	n/a	73.7	73.9	84.3	79.2

Table 1: Frequency of different behaviors in (Baker, Corbett, and Koedinger 2001,2002)

Given this, the total number of parameters fell between 28 and 34 for the different models.[3] It is relevant to note that, by comparison, a prior model of the same phenomena which used ACT-R 4.0-style retrievals (Baker, Corbett, and Koedinger 2002b) used 18 parameters in the model corresponding to our current 34 parameter model. This is because ACT-R 5.0 models are of substantially finer granularity than ACT-R 4.0 models, and suggests that models in the two architectures should not be directly compared using the methods presented here. In the long term, an architecture that compiles directly between different grain-sizes, such as ACT-Simple (Salvucci and Lee 2003), may render this limitation less relevant.

We used the extra-sums-of-squares-F-test and BiC, the Bayesian Information Criterion (Raftery 1995), for our model comparisons. The F-test was used to determine whether there was a statistically significant difference between two of the models which appeared to explain substantially different amounts of the data for situations where one model was a subset of the other, and the BiC was used to compare the relative probability of models where either of these conditions did not hold. In order to use these methods, we needed each model's residuals compared to the original data, computed by subtracting the matrix of model predictions from the matrix of values in the original data (shown in Table 1), and each model's degrees of freedom.

[3] The low ratio between number of proportions in our data and number of free parameters might suggest our models are over-fit, but the proportions are based on the performance of 146 students, and the model's performance could therefore be re-analyzed as the residuals on each of the 3,796 cases. Since this would only affect assessments of the overall quality of the models, the simpler characterization of the data is preferable, being easier to use to compare the two models. Additionally, a low ratio between data set size and free parameters does not negatively affect either of the methods we use in the next section.

Model Comparisons

The first issue we studied through model comparison was whether there was evidence that any of the students in the original study, who had completed a unit of traditional classroom instruction on scatterplots in the previous year, had any understanding of scatterplots at all. We compared a model where some students understood what type of information was used in scatterplots and other students understood how to represent quantitative variables properly (KNOW-IT-ALL) to a model where students understood the information used in scatterplots but knew nothing about quantitative variables outside that context (KNOW-SCATTERPLOTS), and to a model where students did not know anything about scatterplots but knew how to represent quantitative variables properly (KNOW-QUANTITATIVES).

KNOW-IT-ALL achieved an excellent fit to the data set, with an r^2 of 0.972, but despite having fewer parameters, KNOW-QUANTITATIVES achieved an even better fit to the data, with an r^2 of 0.976. Given this it was unsurprising that there was very strong evidence that KNOW-QUANTITATIVES was more probable (BiC=181.8) than KNOW-IT-ALL (BiC=194.1)[4]

KNOW-SCATTERPLOTS achieved substantially poorer fit to the data than either of these models, with an r^2 of 0.916. The difference between KNOW-SCATTERPLOTS and KNOW-IT-ALL was significant, $F(26,1)=659.6$, $p=0.03$, and there was very strong evidence that KNOW-QUANTITATIVES (BiC=181.8) was more probable than KNOW-SCATTERPLOTS (BiC=270.0).

These model comparisons demonstrate that there is no evidence that these students knew anything about scatterplots at all -- the model where no students knew anything about scatterplots was found to be the most probable. On the other hand, if students did not understand quantitative variables in and of themselves, it substantially reduced the model's fit.

A second issue we investigated through model comparison was whether the students used the information given in the question (which implicitly indicated which variable to place on each axis). We compared model KNOW-IT-ALL to a model where students could not use the information given in the question (CAN'T-USE-QUESTION). CAN'T-USE-QUESTION had a considerably worse fit on the surface, with $r^2= 0.79$, and fit the data significantly less well, $F(26,2)=105.1$, $p=0.01$. Hence, our modeling provided evidence that many students were using the information in the question to get correct results.

A third issue we investigated through model comparison was whether the variable choice error and nominalization error stemmed from students randomly choosing variables and randomly choosing how to represent the given variables, or from inappropriate transfer of knowledge of how to choose and represent information in bar graphs. Model KNOW-IT-ALL modeled some students as knowing bar graphs and attempting to create them in the task at hand, whereas model DON'T-KNOW-BAR-GRAPHS eliminated all such skill – hence, any instances of the variable choice error or nominalization error would occur because of random choice

[4] When interpreting values of BiC, the absolute values of BiC for each model are unimportant compared to the values of the models vis-à-vis each other. A difference of more than 6 indicates strong evidence, and more than 10 indicates very strong evidence. (Raftery 1995)

(though the degree of preference for quantitative or nominal variables could still be other than 50/50 at each step). DON'T-KNOW-BAR-GRAPHS had poorer surface fit, with $r^2 = 0.90$, and fit the data significantly less well, $F(26,6)=16.94$, $p<0.0001$. Hence, it seems most likely that student performance was affected by transfer of pre-existing knowledge about bar graphs.

Conclusion

In this paper, we presented a set of techniques for making a principled comparison of the goodness of fit of two computational models without developing closed-form equations for those models. We presented a procedure that treats the computational models as approximations of the closed-form equations which can be derived from them, and showed how to determine a reasonably fair number of free parameters for those models. We then showed how this procedure was used in conducting statistical tests to compare different models of student errors in scatterplot generation.

Acknowledgments

This research was supported by an NDSEG (National Defense Science and Engineering Graduate) Fellowship, by a research contract from Carnegie Learning Inc: "Cognitive Tutors for Middle School Mathematics", and by NSF grant number 9720359 to "CIRCLE: Center for Interdisciplinary Research in Constructive Learning Environments".

We would like to warmly thank Brian Junker, Christian Schunn, and Rhiannon Weaver for helping us refine many of the ideas in this paper, and Christian Lebiere for providing us with the implementation of iterative gradient descent we used to refine our model's predictions. We would like to also thank Benoit Hudson, Samuel Baker, Adam Fass, John Graham, Andrew Ko, Benjamin MacLaren, Hedderik van Rijn, Irina Shklovski, Atsushi Terao, and others for helpful discussions and suggestions.

References

ACT-R Research Group, (2002) ACT-R 5.0 Tutorial Units. http://act- r.psy.cmu.edu/tutorials/

Anderson, J.R. & Lebiere, C. (1998) Knowledge Representation. In Anderson, J.R. & Lebiere, C. (Ed.) Atomic Components of Thought. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J.R. & Matessa, M.P. (1997) A production system theory of serial memory. Psychological Review, 104, 728-748.

Anderson, J.R. & Reder, L.M. (1999) The fan effect: New results and new theories. Journal of Experimental Psychology: General, 128, 186-197.

Baker R.S., Corbett A.T., Koedinger K.R. (2002a) The Resilience of Overgeneralization of Knowledge about Data Representations . Presented at American Educational Research Association Conference. http://www.cs.cmu.edu/~rsbaker/BCKAERA2002.pdf

Baker R.S., Corbett A.T., Koedinger K.R. (2002b) Distinct Errors Arising From a Single Misconception. Published as abstract, Proceedings of the Cognitive Science Society Conference, p. 990, 2002.

Baker R.S., Corbett A.T., Koedinger K.R. (2001) Toward a Model of Learning Data Representations. Proceedings of the Cognitive Science Society Conference, 45-50.

Baker, R.S., Corbett, A.T., Koedinger, K.R., Schneider, M.P. (2003) A Formative Evaluation of a Tutor for Scatterplot Generation: Evidence on Difficulty Factors. To Appear At Conference on Artificial Intelligence in Education.

Gluck, K. A., Pew, R. W. (2002) The AMBR Model Comparison Project: Round III — Modeling Category Learning. Proceedings of the Cognitive Science Society Conference, 24, 19-20.

Koedinger, K.R., MacLaren, B.A. (2002) Developing a Pedagogical Domain Theory of Early Algebra Problem Solving. Technical Report CMU-HCII-02-100, Carnegie Mellon University, Pittsburgh, PA. http://reports-archive.adm.cs.cmu.edu/anon/2002/CMU-CS-02-119.pdf

Lebiere, C., Wallach, D., & West, R. L. (2000). A memory-based account of the prisoner's dilemma and other 2x2 games. In Proceedings of International Conference on Cognitive Modeling, 185-193. NL: Universal Press.

Lebiere, C., & West, R. L. (1999). A dynamic ACT-R model of simple games. In Proceedings of the Twenty-first Conference of the Cognitive Science Society, pp. 296-301. Mahwah, NJ: Erlbaum.

Mitchell, T. (1997) Machine Learning. Boston, MA: WCB/McGraw-Hill.

Nokes, T., Ohlsson, S., and Corrigan-Halpern, A. (2002) Learning by analogy vs learning by instruction: Same knowledge, different representations. Proceedings of the 9th Annual ACT-R Workshop.

Pitt, M.A., Myung, I.J., & Zhang, S. (2002) Toward a method of selecting among computational models of cognition. Psychological Review, 109, 472-491.

Raftery, A.E. (1995) Bayesian Model Selection. Sociological Methodology, 111-196.

Roberts, S. & Pashler, H. (2000) How Persuasive Is a Good Fit? A Comment on Theory Testing. Psychological Review, 107, 2, 358-367.

Salvucci, D. & Anderson, J.R. (1998) Analogy. In Anderson, J.R. & Lebiere, C. (Eds.) The Atomic Components of Thought, 343-384. Mahwah, NJ: Erlbaum.

Salvucci, D.D. & Lee, F.J. (2003) Simple Cognitive Modeling in a Complex Cognitive Architecture. Proceedings of the Association of Computing Machinery Conference on Computer-Human Interaction (CHI 2003), 265-272.

Schunn, C. D. & Wallach, D. (2001) Evaluating Goodness-of-Fit in Comparison of Models to Data. Online Manuscript. http://www.lrdc.pitt.edu/schunn/gof/index.html

Simon, H.A. (1992) What Is an "Explanation" of Behavior? Psychological Science, 3 (3), 150-161.

Stilson, D.W. (1966) Probability and Statistics in Psychological Research and Theory. San Francisco, CA: Holden-Day.

Zucchini, W. (2000) An Introduction to Model Selection. Journal of Mathematical Psychology, 44, 41-61.

Young, R.M. and Cox, A. (2002) Random walk processes in ACT-R mechanisms lead to a wild distribution of learning times. Paper presented at the Eighth Annual ACT-R Workshop, Pittsburgh, PA.