Statistical Techniques for Comparing ACT-R Models of Cognitive Performance

Ryan Shaun Baker, Albert T. Corbett, Kenneth R. Koedinger

Human-Computer Interaction Institute

Carnegie Mellon University

{rsbaker, corbett+, koedinger} @ cmu.edu

http://www.cs.cmu.edu/~pact

With help from Brian Junker and Rhiannon Weaver

Thanks: Chris Schunn, Hedderik van Rijn, Christian Lebiere, NDSEG Fellowship

Why You Should Not Go (Back) To Sleep Right Now

- We will offer an answer to an age-old question
  - "How many times should I run my model?"

Why You Should Not Go (Back) To Sleep Right Now

- We will offer the beginnings of a method to conduct statistical analysis on computational models
  - A method for bounding a model's predictions
  - Propose a "new" parameter counting strategy to assess the relative complexity (and potential for overfitting) of different models
  - A reasonable strategy for model selection given these components

Lecture Map

- The Challenges to Model Comparison
- Bounding A Model's Predictions
- Assessing Model Complexity
- Model Comparison
- Conclusions

The Reason For Model Comparison

- We have two potential models for a single phenomenon
  - Which one better explains the data?
  - Increasingly important to the ACT-R community
  - Recent examples
    - Lovett 2000
    - Taatgen & Anderson 2002
    - Byrne 2001
    - Fum and Stocco 2002

Two (Broad) Approaches to Comparing Models

1. Find a new data set and see how well they generalize
   - Gluck and Pew (2002)'s AMBR competition

Two (Broad) Approaches to Comparing Models

2. Compare models within original data set
   - Roberts and Pashler (2000) argue that computational modelers do not pay proper attention to the number of *free parameters* (quantities allowed to vary during model fitting)
   - There's some fairness in this criticism. A lot of attention is paid to goodness-of-fit, but is the model better or just massively overfit?
   - The statistical community has model selection formulas that take this into account (Zucchini 2000, Pitt et al 2002)

The Problem With Traditional Model-Selection Formulas

- But these model selection formulas generally can't be used without closed-form equations because it is difficult to
  - Determine the proper number of free parameters
    (quantities affecting the results that are allowed to vary)
  - Get exact model predictions

Develop Closed-Form Equations?

- Some ACT-R modelers have responded to this need by developing closed-form equations describing their model's behavior
  - Anderson and Matessa 1997
  - Koedinger and MacLaren 2002

Develop Closed-Form Equations?

- But this is extremely difficult for even moderately complex ACT-R models (Schunn and Wallach online)
- Koedinger and MacLaren's (2002) model requires 55 equations with as many as 44 terms in a single equation

Another Solution...

- We propose a different solution…
- Instead of developing the equations,
- We will run the computational model
  - And run it long enough to get a good approximation of the mathematical model
  - And come up with a reasonably principled parameter count
  - And then we will pretend we have a mathematical model
  - And use model selection methods from the statistical community

Terminal Models

- For tractability, we will generalize this discussion only to "terminal models" of cognitive performance (Salvucci and Anderson 1997)
- Involve learning of new chunks, but not learning of new productions or changes in production weights
- The methods presented here can definitely be generalized to models with full production learning

Lecture Map

- The Challenges to Model Comparison
- Bounding a Model's Predictions
- Assessing Model Complexity
- Model Comparison
- Conclusions

Approximating the Mathematical Model

- 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 the model a greater number of times (from the same initial state or set of initial states), a more accurate approximation of the solution of those equations is developed
- When the model is run an infinite number of times, the error of the approximations will reach 0

How many times should we run the model?

We can make sure

each of the model's results fall

within a certain confidence interval

of what the mathematical model would give

a pre-selected percentage of the time.

Let's Say…

- Let's say we want every prediction in the computational model
  - To fall within 5% of the prediction the mathematical model would make
  - 95% of the time
  - Given a worst-case standard deviation
- Then…

Standard Equation for Confidence Intervals

Found in your favorite elementary stats textbook – mine is Rosenthal & Rosnow (1991)

The distance each result is allowed to fall from the value the mathematical model would give

The number of runs

$$d = (t_n(\alpha/2)) * s / \sqrt{n}$$

The t-distribution value for n runs and α proportion of the time

(wondering why the t-distribution is ok? Ask me during the question period!)

The proportion of the time you're willing to let each result be more than d away from the result the mathematical model would give

The standard deviation of each computational run from the result the mathematical model would give

Standard Equation for Confidence Intervals

Or, by algebraic transformation…

$$n = ((t_n(\alpha/2)) * s / d)^2$$

Computing…

$n = ((t_n(0.05/2)) * \sqrt{0.5} / 0.05)^2$

Via algebra…

n = 778

N = 778

- n=778 is a magic number.
- It's more principled than n=100 or n=1000 (or n=1), but it's still arbitrary
- It would vary
  - With a more liberal estimate of deviation
  - If you wanted bounds within 2% or 10% rather than 5%
  - If you wanted to bound the variance of *all* predictions rather than *each* prediction
- Nonetheless, I recommend that everyone
  - Use n=778
  - Cite me liberally and sometimes inappropriately

Lecture Map

- The Challenges to Model Comparison
- Bounding A Model's Predictions
- Assessing Model Complexity
- Model Comparison
- Conclusions

A Rather Simple Heuristic

- In this talk, we'll
  - propose a "new" parameter counting strategy for counting free parameters, to help us assess model complexity
  - offer informal evidence in favor of this heuristic

Free Parameter Count

- Like many ACT-R modelers
  - Each ACT-R global parameter you allow to vary = 1 free parameter
- But also
  - Each production that affects the results of the model = 1 free parameter
  - Each declarative chunk that affects the results of the model = 1 free parameter

The Key to Counting Free Parameters

- What factors influence the mathematical equations that describe the cognitive model?
- Some approaches to model selection, like MDL (Pitt et al 2002), take the relative influence of different factors into account
- More approximative model selection methods, like BiC (Raftery 1995), treat all factors the same but want to know how many there are

  (which is what we're doing here)

The mathematical equations underlying ACT-R

- For any ACT-R model, the equations that underlie its behavior will be some function of the equations underlying ACT-R

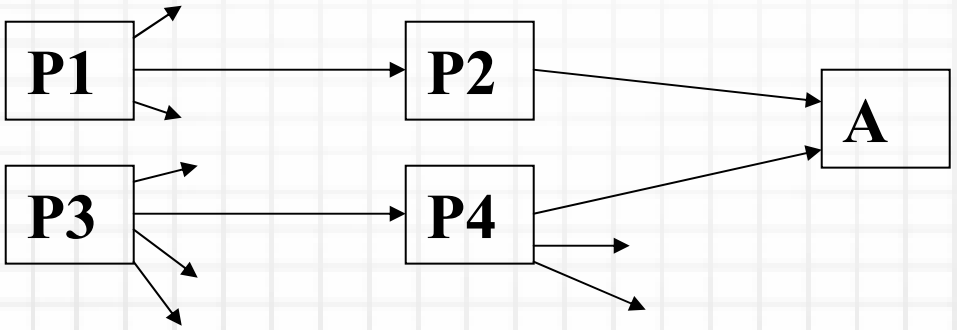
The Probability of a Specific Behavior

- P (Behavior) = Σ P(all production chains producing behavior)

- P (chain) = Π P(each production in chain)

The Probability of a Specific Behavior

- So if behavior A is produced by production P1 followed by P2 or P3 followed by P4, then
- P(A)=P(P1)*P(P2| P1) + P(P3)*P(P4| P3)



Probability of a specific production firing

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

U_i ← The expected utility of production Pi

U_j ← The expected utility of each of the productions that could fire at this point

s ← Noise variance

The probability of behavior A

- Is a function not only of the utility of productions $P_{1\text{-}4}$ that can fire to produce behavior A
- But also the utility of each production $P_{5\text{-}\infty}$ that *could have fired,* preventing productions $P_{1\text{-}4}$ from firing
- So every production that fires or could fire during the model's execution *affects the equations* and should be counted as at least one free parameter
- There are some exceptions
  - Productions "yoked together"
  - Productions that fire in every run

Chunks

- What about chunks?
- The same analysis applies, more or less…

Chunks

- The probability of retrieving a given chunk i, is:

$$P(i) = \frac{e^{\frac{A_i}{\sqrt{2s}}}}{\sum_j e^{\frac{A_j}{\sqrt{2s}}}}$$

The activation of chunk Ci

The activation of each of the chunks that could be retrieved at this point

The Formula for Spreading Activation

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

Base-level activation of chunk Ci

Spreading activation from other chunks which have references to chunk Ci

So…

- As with productions, count as a parameter
  - Each chunk that could be retrieved
- But also
  - Each chunk that spreads activation to a chunk that could be retrieved
- Because both affect the resultant equations
- (With similar caveats as before)

What about chunks learned?

- Count them too
  - Each unique chunk C_* created during any run which can be retrieved or can spread activation will affect your model's equations
  - C_*'s existence or non-existence during any given run will definitely affect the results of the run
- There are some considerations of what's truly a "unique chunk"
  - When one can be retrieved and the other can't…
- This could be very easily extended to production learning…

ACT-R Global Parameters

- Any ACT-R global parameter allowed to vary will certainly affect the results of the equations
  - Possibly more than any individual production or chunk
  - So count each ACT-R global parameter you allow to vary
    - I believe there's consensus on this, so I won't go into why you don't need to count the other ones…

Free Parameter Count -- Summary

- With the clarifications noted before
  - Each production that affects the results of the model = 1 free parameter
  - Each declarative chunk that affects the results of the model = 1 free parameter
  - Each ACT-R global parameter you allow to vary = 1 free parameter

This can't be used to compare ACT-R 4 and ACT-R 5 models

- ACT-R 5 models usually (and unsurprisingly) have substantially more productions than ACT-R 4 models.
- The parameter-counting method we use here only makes sense for productions of approx. equal grain size
- But this is well upheld in ACT-R 5!
- And in the long-term, architectures like ACT-SIMPLE (Salvucci & Lee 2003), which directly compile between models of different grain sizes, may solve this problem

Lecture Map

- The Challenges to Model Comparison
- Bounding A Model's Predictions
- Assessing Model Complexity
- Model Comparison
- Conclusions

Context of Use

- We were about to design a Cognitive Tutor (Anderson et al 1995) for Scatterplot Generation
- We wanted a good model of the behavior students displayed while performing this task
- 2 studies with 5 conditions showed 2 frequent errors, which looked similar. Were they similar at a deeper level as well?
- What skills underlied the pattern of correct performance that did occur?

Comparing Model Variants

- We compared 5 model variants to one another to determine which knowledge components corresponded to a better explanation of the data
- This helped us to understand the sources of student errors and successes and design a better tutor

Finding Each Variant's Predictions

- We ran Lebiere's implementation of Iterative Gradient Descent (IGD) on each model variant, minimizing a function of r^2 and Mean Absolute Deviation (MAD)
- For each run during IGD and for final predictions, we ran the model 778 times.

Statistical Techniques Used

- Extra-Sum-of-Squares F-tests
- Bayesian Information Criterion (BiC)
- Both require
  - A set of predictions for each model
  - A parameter count for each model
  - Which we have now!
- Both give
  - Assessments of which model (among 2) explains the data better
  - Assessment of whether the difference is statistically significant

Lecture Map

- The Challenges to Model Comparison
- Bounding A Model's Predictions
- Assessing Model Complexity
- Model Comparison
- Conclusions

What we have now

- The beginnings of a method to conduct statistical analysis on computational models
  - A method for deciding how many times to run a model to get known bounds on its predictions
  - Some hand-waving proof that a fairly simple strategy for counting parameters is an appropriate way to assess model complexity
  - A reasonable strategy for model selection given these components

Where we're going next

- Come up with a better way of finding each model variant's parameter values
  - More verifiably MLE than IGD to minimize MAD/r^2
  - Petrov (2001) offers valuable suggestions for doing this
- Come up with a formal method for relating the uncertainty in the computational model's predictions to the uncertainty quantified by F tests and BiC
  - What we're doing is probably valid but needs to be formalized
- Verify (or, if necessary, refine) our parameter-counting strategy
- Offer a package, in LISP, to assist others in conducting these analyses

Acknowledgements

Brian Junker
Christian Schunn
Christian Lebiere
Andrew Ko
Benjamin MacLaren
Adam Fass
Samuel Baker
Rhiannon Weaver
Benoit Hudson
John Graham
Hedderik van Rijn
Irina Shklovski
Atsushi Terao

The End

- Questions?

Productions You Don't Need To Count

- Any production $P_{tagalong}$ that always fires when production P_1 has already fired
  - Only true if it always fires… if nothing fires due to utility threshhold, this can't be applied
- Productions which fire in every single run
  - For book-keeping or initial goal-setting
  - Each declarative chunk that affects the results of the model = 1 free parameter
  - Each ACT-R global parameter you allow to vary = 1 free parameter

Each Production's Utility

- $U_i = \pi_i G - C_i + \varepsilon$

U_i	Production i's utility
π_i	The expected probability that choosing production i will lead to the desired result
G	The current objective's value
C_i	The expected cost of firing the production
ε	Noise parameter

If you allow both π_i and C_i to vary, you may need to count two global parameters…

Chunks You Don't Need To Count

- Any Chunk C_{info} used solely for information storage
  - C_{info}'s retrieval never fails
  - There is never a case where C_{info} is competing with another chunk for retrieval
  - C_{info} spreads no activation
- Any chunk $C_{tagalong}$ which is always retrieved after a specific production has already fired

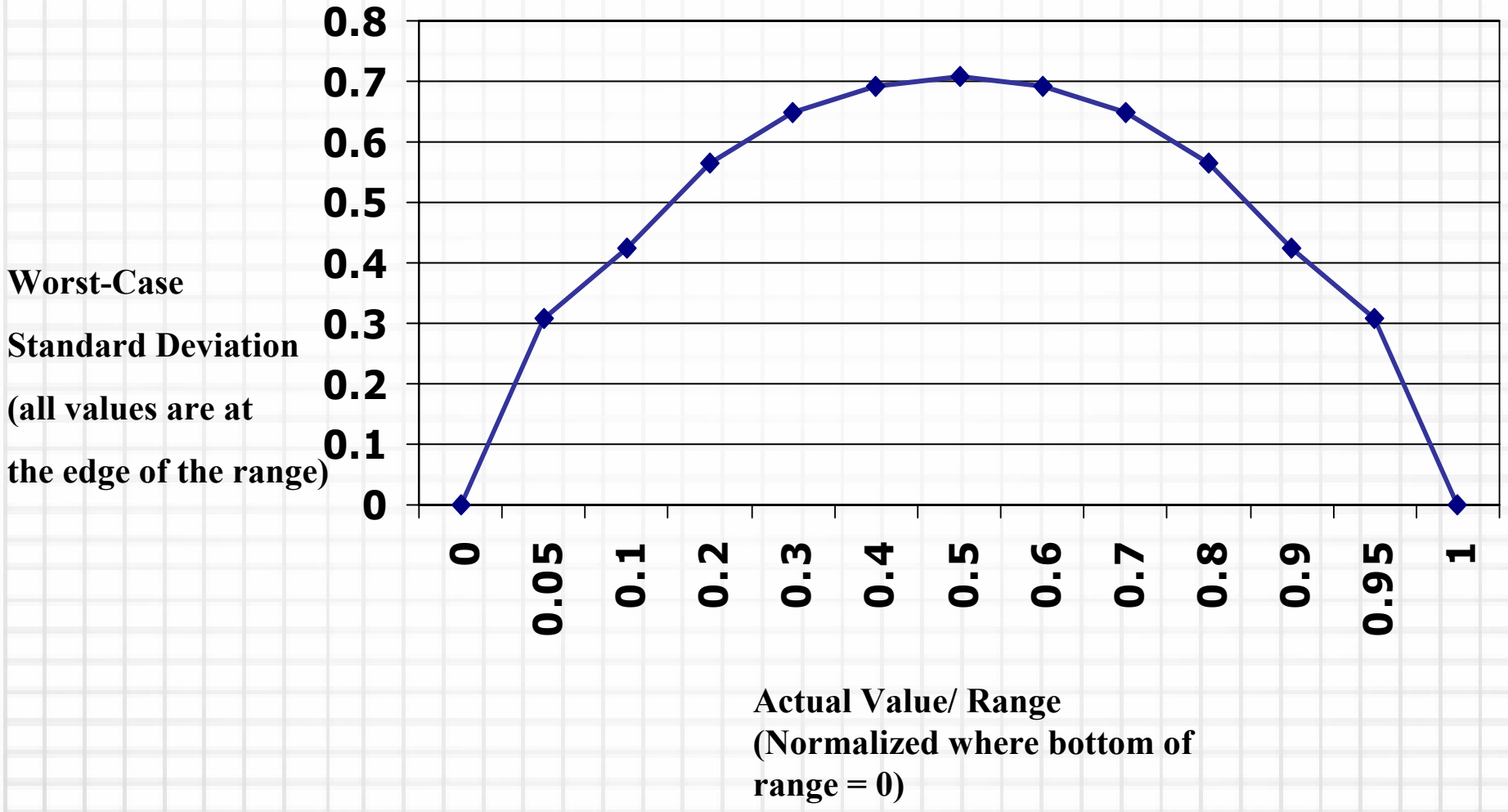
Standard-Deviation Assumption

- What is the worst-case assumption for the standard deviation of each individual run from the actual proportion?
- This is when every individual run gives a result that's ½ the range of possible values of data away from the actual value.
- For example, if the results are expressed as frequencies between 0 and 1, then the worst case for data concerning proportions of events is when the actual proportion is 0.5. (and each event is 0 or 1.)
- So the worst-case standard deviation is $\sqrt{0.5}$

Global Parameters You Don't Need To Count

- If you never allow a global parameter to vary
  - If you leave it at 0, or ACT-R default, or a well-known value from a previous experiment (as in Lebiere and West, 1999 and Lebiere, Wallach, and West, 2000)
- Then it can be treated as a constant in the equations and not counted as a free parameter
- If you tweaked it, or tried different values, count it as a global parameter…
- This calls for honesty and clear reporting on the part of the modeler!

Standard-Deviation Assumption



T-Distribution Assumption

- The t-distribution will be an appropriate predictor of the deviation of each model run from the actual proportion
- Not valid for all ACT-R models (cf Young and Cox 2002), but transformations can be used
- And for large enough numbers of runs, Central Limit Theorem guarantees normal distribution will be a valid assumption

Terminal Models

- Terminal Models
  - Involve learning of new chunks, but not learning of new productions or changes in production weights
  - Are valid to use "in cases where the behavior under study is at some relatively asymptotic level or the critical factors being investigated do not change over the range of experiences encountered in the experiment." (Anderson, Lebiere and Lovett 1998)
  - "a typical assumption in much of the experimental research on human cognition"
- Many Recent Terminal Models:
  - Tower of Hanoi (Anderson & Lebiere 1998)
  - Fan Effect (Anderson & Reder 1999)
  - Student Thinking After Instruction (Nokes et al 2002)

Model Variant Comparisons

- Model KNOW-QUANTITATIVES
  - Students know how to represent quantitative variables properly
  - $r^2 = 0.976$

    BiC = 181.8
- Model KNOW-SCATTERPLOTS
  - Students do not know that but
  - Students do know that Scatterplots contain quantitative variables
  - $r^2 = 0.916$
  - BiC = 270.0
- "Very Strong" evidence in favor of KNOW-QUANTITATIVES

Model Variant Comparisons

- Model KNOW-IT-ALL
  - Students can (and do) read the question to help them figure out what variables to use
  - $r^2 = 0.972$
- Model CAN'T-USE-QUESTION
  - $r^2 = 0.79$
- $F(26,2)=101.5$, $p=0.01$

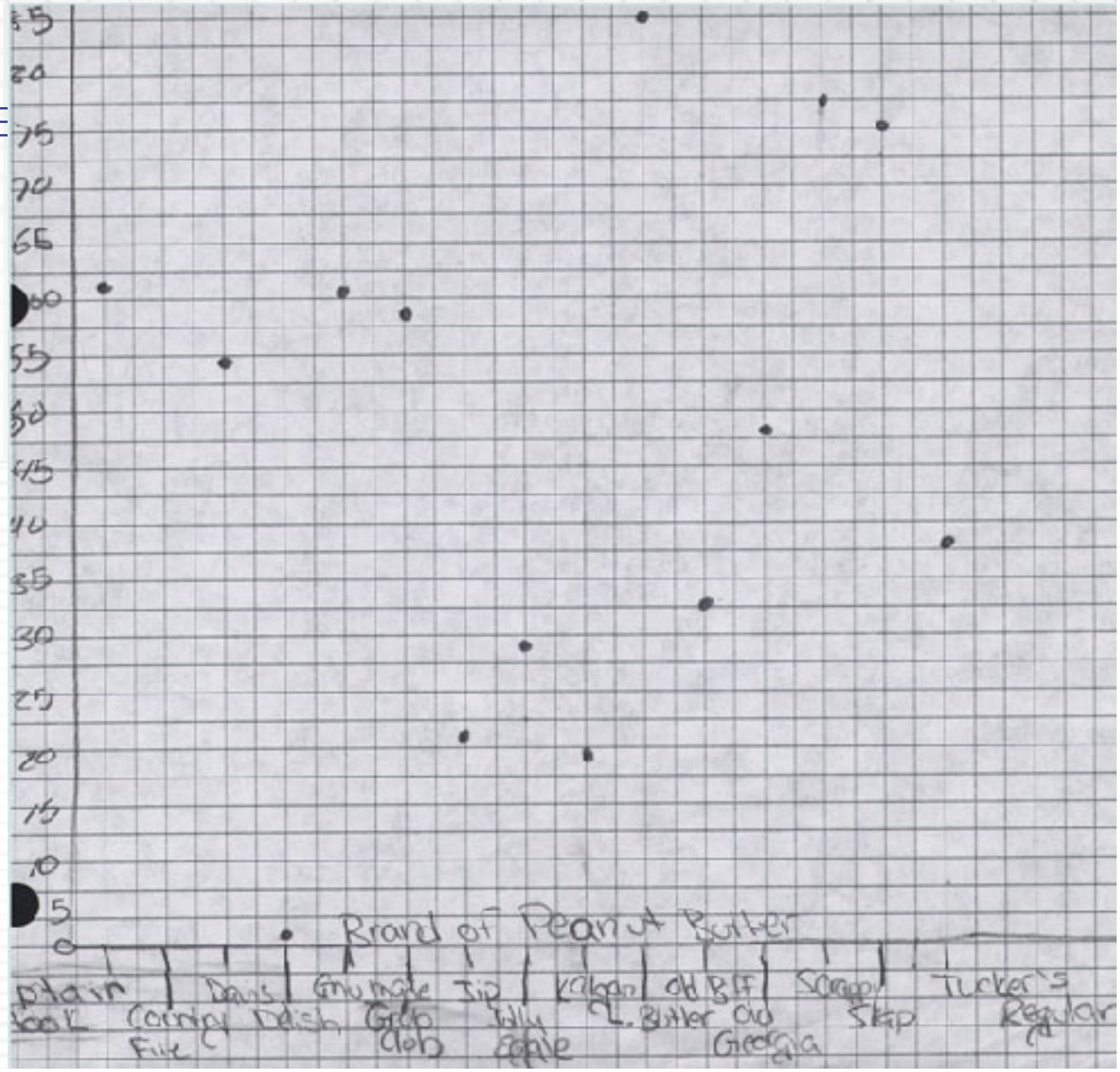
Student Errors in Scatterplot Generation

- Two errors where the student ends up with a nominal X variable, which is appropriate for bar graphs but not for scatterplots
  - The "Variable Choice" Error
    - The student places a nominal variable on the X axis
  - The "Nominalization" Error
    - The student places the correct quantitative variable on the X axis but treats it as if it were nominal

"Variable Choice" Error

Students asked to draw scatterplots

Used variables more appropriate for bar graphs, placing a nominal variable on the X axis rather than a quantitative one.



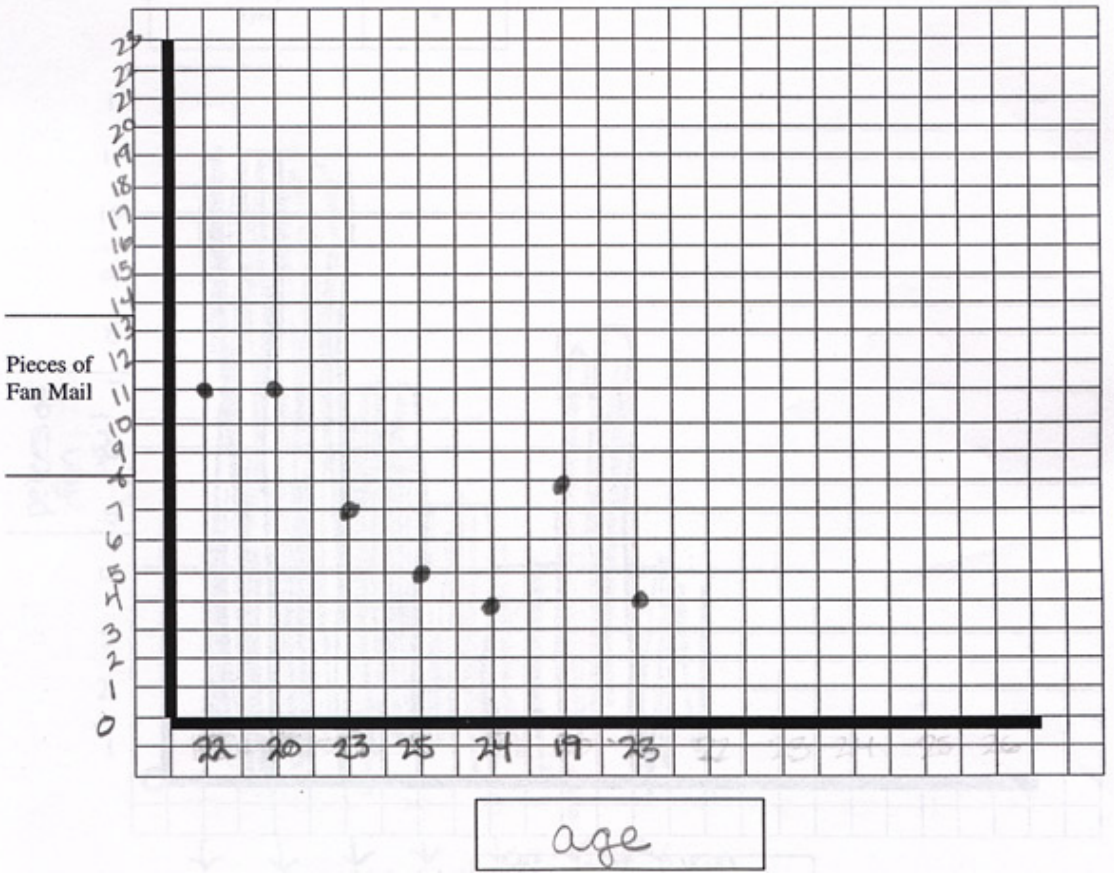
"Nominalization" Error

Students asked to draw scatterplots

Used the correct variables, but turned the X axis variable into a nominal variable, as if it were a bar graph.

This data shows the ages of several musicians and the number of pieces of fan mail they receive each day, in thousands.
Please draw a scatterplot, to show if the amount of fan mail a musician gets is related to their age.
Show all work.
Hint: Scatterplots are made up of dots.

Musician	Age	Pieces of fan mail (thousands)
CJ	22	11
Nick	20	11
Devin	23	7
Glennis	25	5
Howie	24	4
Britney	19	8
Ryan	23	4



Data Set Features

- 5 conditions in the original studies (between-subjects)
- Different proportion of correct behavior and each error, as well as other behaviors, in each condition

	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)

Parameters

- 6 declarative chunks
- 3 ACT-R global parameters
- Between 19 and 25 productions in the different variants
- 28 to 34 parameters total

Model Variant Comparisons

- Model KNOW-IT-ALL
  - Students' prior knowledge of bar graphs influences some of them to mis-transfer bar graph features into their scatterplots
  - r^2= 0.972, 34 parameters
- Model DON'T-KNOW-BAR-GRAPHS
  - Any student behavior that mimics bar graph features arises solely through random chance
  - r^2= 0.90, 28 parameters
- $F(26,6)=16.94$, $p<0.0001$

Model Variant Comparisons

- Model KNOW-QUANTITATIVES
  - Students know how to represent quantitative variables properly
  - r^2= 0.976, 29 parameters

    BiC = 181.8
- Model KNOW-IT-ALL
  - Students know that, and
  - Students know that Scatterplots contain quantitative variables
  - r^2= 0.972, *and* 32 parameters
  - BiC = 194.1
- BiC Difference=12.3, "Very Strong" evidence in favor of KNOW-QUANTITATIVES

  Smaller BiC values are better. A difference of 6 is "Strong", a difference of 10 is "Very Strong" – (Raftery 1995)

Conclusions About Student Cognition

- Our modeling suggested that the pattern of errors students made
  - Stems from understanding what kinds of variables go into bar graphs, and mis-transfering this knowledge to scatterplots
- Our modeling suggested that the correct performance we saw
  - Did not stem from understanding scatterplots
  - Instead, stemmed from a combination of
    - Understanding what a quantitative variable is
    - Knowing how to use the information given in the question
- This had implications for the design of our tutor. (Baker et al 2003)

Some Future Work…

- We currently give one BiC value (and one F-value), but given the stochastic nature of the model predictions, it might be better to give a range of values that BiC could be, or to adjust our F-values downwards
- We’re looking into this

How Tight Should Predictions Be?

- Look at the confidence intervals in the original data set
- If you make your bounds substantially tighter than these
  - you may spend time during the parameter-fitting stage getting a tighter fit to the error in the original data set.
  - you may end up choosing the model which best fits the error in the original data set

Model Complexity

- The more complex we allow a model to be, the more likely
  - it can fit an arbitrary data set just by chance
  - it will overfit by going beyond fitting the data, by fitting the error in the data set
- So it's important to assess the trade-off between
  - how closely a model fits the data (goodness-of-fit)
  - and how complex the model is